

Системы учебных задач для обучения алгоритмизации в пропедевтическом школьном курсе информатики

Н.Д. Шумилина

С первых лет введения курса школьной информатики в старших классах более 20 лет тому назад апробировался и развивался курс раннего обучения информатике. Однако за прошедшие годы Государственный стандарт начального общего образования по информатике не был принят. Кроме того, отсутствие единого, цельного, непрерывного школьного курса информатики в федеральном базовом учебном плане говорит о проблеме, которая ждёт своего решения. Информатика служит целенаправленному развитию операционного стиля мышления человека — фундамента освоения любой технологии, что должно во-зобладать над идеями быст-

рого и легковесного «научения компьютеру», а непрерывный школьный курс информатики должен найти своё место в учебном плане российских школ.

Современный операционный стиль мышления включает в себя (с некоторой долей условности) несколько составляющих: планирование, моделирование, поиск, общение, инструментирование, о чём сказано в работе Ю.А. Первина¹. Развитие умения планировать структуру действий определяется в первую очередь алгоритмическими разделами курса. При этом влияние планирования на остальные составляющие, безусловно, велико.

Содержание существующих курсов раннего обучения информатике имеет отличия во введении отдельных понятий, в трактовках, подходах. Даже для общепризнанных элементов обучения, например, управляющих структур, не существует достаточных разработок, обеспечивающих качественное обучение, позволяющее школьникам освоить ту или иную тему, осознанно применять полученные знания при изучении других предметов и на практике.

Назрела необходимость в системной, методически и научно обоснованной разработке содержания обучения алгоритмизации в курсе раннего обучения информатике. Содержание обучения алгоритмизации должно быть обосновано с точки зрения введения базовых понятий, структур, порядка их освоения, разработки соответствующих систем учебных задач.

¹ Первин Ю.А. Методика раннего обучения информатике: Методическое пособие. М.: БИНОМ. Лаборатория знаний, 2005. С. 15.

Отметим полное согласие авторов широко известных курсов раннего обучения информатике в необходимости освоения основных алгоритмических структур. Таким образом, необходима подборка задач, обеспечивающая качественное и контролируемое освоение основных алгоритмических конструкций. В работе В.В. Гузеева² описывается так называемый задачный подход к обучению, в котором планируемые результаты обучения представляются не просто множествами, а системами учебных задач. К системе задач автор относит совокупность задач к блоку уроков по изучаемой теме, удовлетворяющую ряду требований, пояснение к которым приведём в сокращённых формулировках.

Полнота. Наличие задач на все понятия и способы деятельности.

Наличие ключевых задач. Наличие задач, имеющих принципиальное значение для усвоения содержания.

Связность. Представление совокупности задач в виде связанного графа, в узлах которого ключевые задачи, выше — подготовительные, ниже — следствия, обобщения.

Возрастание трудности в каждом уровне. Предлагается три уровня: минимальный, общий и продвинутый уровни планируемых результатов обучения. В каждом из уровней трудность задач непрерывно нарастает.

Целевая ориентация. Для каждой задачи определено её место и назначение в блоке уроков.

Целевая достаточность. В системе достаточно задач для тренажа в классе и дома, для индивидуальных и групповых заданий разной направленности, для самостоятельной деятельности учащихся, для текущего и итогового контроля.

Психологическая комфортность. Наличие различных заданий с учётом разных темпераментов, типов мышления, видов памяти: устных и письменных упражнений, словесных, графических, предметно-иллюстративных представлений заданий и решений, задач-шуток и т.д.

В работе И.Г. Семакина, Т.Ю. Шеиной³ говорится о том, что последовательность задач для освоения алгоритмических разделов должна удовлетворять следующим характеристикам.

От простого к сложному. Постепенное усложнение задач.

Новизна. Добавление элементов знания с каждой новой задачей.

Наследование. Необходимость использовать знания, полученные при решении предыдущих задач.

Очевидно, что требования, изложенные в работе В.В. Гузеева, полнее предъявленных характеристик, поскольку понятия новизны и наследования включаются в понятия полноты и связности. Кроме того, ориентация на планируемые уровни обучения конкретизирует систему задач, её границы и уровни.

В разработанных системах задач для начального обучения алгоритми-

² Гузеев В.В. Эффективные образовательные технологии: интегральная и ТОГИС. М.: НИИ школьных технологий, 2006. С. 17.

³ Семакин И.Г., Шеина Т.Ю. Преподавание базового курса информатики в средней школе: Методическое пособие. М.: БИНОМ. Лаборатория знаний, 2006. С. 214.

ТЕОРИЯ ДЛЯ ПРАКТИКОВ

зации⁴ выделялись характеристики, близкие к сформулированным В.В. Гузеевым. Однако отметим три отличия, которые были допущены при разработке систем задач для обучения алгоритмизации младших школьников.

В формулировке связности задач необходимо показать связанность уровней. Минимальный — подготовительный к общему уровню, на котором должны находиться ключевые задачи. В нашем случае ими можно считать задачи, запланированные для контроля усвоения изученного материала. Они становятся ориентиром усвоения той или иной структуры.

В системах задач обучения алгоритмике речь идёт о сложности как объективной характеристике, не зависящей от индивидуальных особенностей и подготовки конкретного ученика. Построение задач на освоение алгоритмических конструкций предполагает постепенное увеличение количества шагов, действий, команд, необходимых для составления алгоритма (в некотором смысле аналогично построению шахматных этюдов — например, в два, три или более ходов).

Необходим учёт возрастного соответствия. Формулировки задач для детей младшего возраста должны быть им доступны и понятны. Поня-

тый словарь младших школьников формируется и необходимо возрастное соответствие используемых в условиях задач понятий и представлений. Психологическая комфортность, безусловно, возможна только при понимании условия задачи. Она представляет собой более высокий уровень требований, предъявляемых к качеству системы задач, следующий после доступности.

При разработке системы задач для каждой алгоритмической конструкции в качестве основы использовались старинные классические сюжеты: о переправах, переливаниях, фальшивых монетах, сортировке поездов, ханойских башнях. С одной стороны, школьники знакомятся с этими сюжетами, с другой — каждая система задач, сформированная в рамках перечисленных требований, будет служить целенаправленному формированию алгоритмического мышления школьников.

Таким образом, системообразующим фактором для всего множества задач является тематическая принадлежность и сложность задачи. Сложность заданий, которые можно сформировать на основе разработанных систем задач, будет зависеть от трёх составляющих: тематической, содержательной и деятельностной. Рассмотрим каждую из них.

⁴ Шумилина Н.Д. Переправа, переправа... : Задачи разного уровня сложности // Информатика в школе: Прил. к журналу «Информатика и образование». 2003. № 6. С. 78–111; Шумилина Н.Д. Переливашка и Водолей : Задачи разного уровня сложности // Информатика в школе: Прил. к журналу «Информатика и образование». 2004. № 3. С. 49–89; Шумилина Н.Д. Машинист... : Задачи разного уровня сложности // Информатика в школе: Прил. к журналу «Информатика и образование». 2004. № 7. С. 37–54; Шумилина Н.Д. Исполнитель Автомат. Задачи разного уровня сложности. Большой Московский семинар по методике раннего обучения информатике ИТО-РОИ-2006/ Публикации <http://ito.edu.ru/sp/publi/publi-0-ShumilinaND.html>.

Тематическая сложность

Предлагаемая последовательность изучения конструкций согласуется с изложением в курсе, разработанном под руководством И.Г. Семкина, а также в работе А.Г. Кушниренко и Г.В. Лебедева⁵, однако для пропедевтического курса внимание должно быть сосредоточено на *формировании* понимания, представления сути основных управляющих конструкций (следование, повторение, ветвление), механизма вспомогательного алгоритма, рекурсии. Применение алгоритмических конструкций для составления алгоритмов должно следовать за первичным этапом освоения перечисленных структур.

Поэтому предлагается частнометодический принцип формирования базовых конструкций из представления последовательности действий. Конструкции следования, вспомогательного алгоритма, повторения (заданное количество раз) вытекают из приоритета понятия следования, которое заложено в самом понятии алгоритма. С ним априори, неявно согласны все авторы предлагаемых методик и курсов, поскольку линейная последовательность действий стоит во всех курсах на первом месте.

Мысль о том, что переход от (статической) программы к (динамическим) вычислениям, в сущности, предоставлен машине», высказал Дейкстра⁶. Он обратил внимание на необходимость понимания разницы пред-

ставления записи и реального выполнения программы (алгоритма).

Предлагается строить обучение индуктивно, показывая, как реально возникает та или иная свёртка линейной (или линейной с пропусками, в случае ветвления) последовательности команд в известные конструкции, а не дедуктивно, объясняя, что существуют изначально заданные конструкции, которые можно развернуть в соответствующие последовательности действий.

Это означает, что:

- необходимо зафиксировать линейную (покомандную) запись алгоритма;
- необходимо организовать переход от объёмной линейной записи к более короткой, с использованием объединения нескольких команд в одну, свёрнутую, т.е. продемонстрировать появление вспомогательного алгоритма;
- необходимо организовать переход от объёмной линейной записи с явными повторяющимися группами команд к более короткой циклической структуре;
- ветвящуюся структуру представлять текстом, состоящим из нескольких строк, читаемым последовательно, с возможными пропусками некоторых команд. В этом проявляется дополнительная сложность алгоритма с ветвлением. И поэтому алгоритмы, содержащие ветвление, имеет смысл рассматривать после вспомогательных и циклических (с заданным количеством повторений) алгоритмов.

⁵ Кушниренко А.Г., Лебедев Г.В. 12 лекций о том, для чего нужен школьный курс информатики и как его преподавать: Методическое пособие. М.: Лаборатория Базовых Знаний, 2000.

⁶ Дал У., Дейкстра Э., Хоор К. Структурное программирование: Математическое обеспечение ЭВМ: Пер. с англ. М.: МИР, 1975. С. 24.

ТЕОРИЯ ДЛЯ ПРАКТИКОВ

Такова логика освоения основных алгоритмических конструкций при учёте общедидактического принципа доступности и специфической реализации его в частнометодическом *принципе формирования базовых конструкций из представления последовательности действий*.

Различия в возможностях построения учебного плана школ требуют составления как минимум двух вариантов обучения, например, базового и расширенного. Расширение достигается за счёт второго «витка» изучения тех же тем на более высоком уровне, а также изучения рекурсии. Концентрическое построение соответствует большим временным возможностям. Таким образом, учитель может выбрать из предлагаемых вариантов обучения тот, который ему больше подходит в зависимости от имеющегося учебного времени, а также возможностей детей. Поэтому ориентировочная основа для составления учебного плана будет следующей:

Базовый вариант.

1. Следование (Переправы);
2. Вспомогательный алгоритм (Машинист 1);
3. Цикл N раз (Повторяйка);
4. Ветвление (Фальшивая монета 1).

Расширенный вариант:

1. Следование (Переправы);
2. Вспомогательный алгоритм (Машинист 1);
3. Цикл N раз (Повторяйка);
4. Ветвление (Фальшивая монета 1);
5. Следование (Пелеливашка);
6. Вспомогательный алгоритм (Машинист 2);
7. Условный цикл (Автомат);
8. Условный цикл, ветвление (Фальшивая монета 2).
9. Рекурсия (Ханойские башни).

Содержательная сложность

Содержательная сложность заданий определяется количеством шагов в линейной записи, количеством выделяемых блоков (вспомогательных алгоритмов, повторяющихся действий) при освоении следования, вспомогательного алгоритма и цикла. Для ветвления сложность определяется деревом возможных состояний объектов задачи, для рекурсии — количеством обращений процедуры к самой себе. На начальном этапе обучение необходимо строить на основе рассмотрения всех вариантов в ветвящейся структуре и развертки рекурсивной структуры до элементарных шагов.

Каждая из тематических систем задач строится на трёх уровнях содержательной сложности, это обеспечивает соответствие основным дидактическим принципам. Задачи разработаны на уровнях:

- ознакомления,
- усвоения,
- повышенной сложности.

Отметим, что начальные, простые задачи служат для формирования умения записи алгоритма. В этом важность задач первого, минимального уровня. Кроме того, следует учесть возможную проблему записи для первоклассников.

На основном уровне фиксируются задачи для контроля усвоения. Именно эти задачи нужно считать ключевыми. Выполнение их свидетельствует об усвоении детьми той или иной темы. Формальное проведение контрольных работ необязательно: решение ключевых задач можно рассматривать как этап закрепления полученных знаний и умений.

Деятельностная сложность

Деятельностная сложность задания может быть отнесена к одной и той же задаче и связана с вариативностью предъявления условия:

- выполнить заданный алгоритм;
- расставить приведённые команды алгоритма по порядку;
- обработать (дописать, исправить, сравнить, построить обратный) алгоритм;
- составить, разработать алгоритм.

В рамках определённой системы задач задания различной деятельностной сложности учитель предлагает, ориентируясь на уровень класса и на

возможности и достижения отдельных детей. Подобным образом разработанные системы задач способствуют и обеспечивают:

- пропедевтику основных алгоритмических структур;
- индивидуальный подход к обучаемым;
- использование задач различного уровня сложности для работ соответствующей целевой ориентации;
- контролируемый (по уровню сложности) процесс обучения;
- включение учащихся в процесс конструирования задач;
- открытость систем для формирования новых задач.