***Т.А. Трохова**, к.т.н., доцент кафедры «Информационные технологии» Гомельского государственного технического университета имени П.О. Сухого,*
***М.Л. Шишаков**, к.т.н., доцент кафедры «Микропроцессорная техника» Белорусского государственного университета транспорта,*
***Е.В. Коробейникова**, ассистент кафедры «Информационные технологии» Гомельского государственного технического университета имени П.О. Сухого*

Системный подход к обучению алгоритмизации в курсе «Информатика»

Тенденции и направления преподавания информатики в непрерывной обучающей среде «школа-вуз», с точки зрения авторов, говорят о том, что доля такой важной темы, как алгоритмизация задачи, в общем объеме дисциплины значительно снижена. Основной упор сделан на изучение систем программного обеспечения компьютера, как то: операционная система, системы, входящие в состав MS Office, системы сетевых технологий и т.д. Достаточное место в некоторых образовательных программах занимает тема непосредственного программирования, которая как бы априори подразумевает и алгоритмизацию: без алгоритма программы не напишешь. Но как строится этот алгоритм, какие правила и рекомендации следует учитывать при его разработке, как выработать у обучаемых алгоритмический подход к решению любой прикладной задачи – подробные ответы на эти вопросы не так часто встречаются в учебных программах курса «Информатика». В итоге после двухлетнего обучения информатике в школе и такого же по времени курса в вузе у студента технической и непрофильной с компьютерами специальности не всегда появляются навыки системного алгоритмического мышления, которые так ему необходимы в дальнейшей деятельности наряду со знанием приемов работы с программным обеспечением компьютера.

В настоящее время, когда средняя школа практически не рассматривает приемов синтеза алгоритмов, вся нагрузка на изучение этой темы падает на вуз. Опыт преподавания информатики в высшей школе позволяет сделать вывод о том, что временные рамки курса не дают возможности уделить алгоритмизации должного внимания, поэтому данная тема рассматривается параллельно с изучением приемов программирования на

каком-либо языке программирования. В некоторой мере это оправдано, так как язык программирования является естественным реализатором алгоритма и наглядно демонстрирует получение конкретных результатов его работы. Но даже при замене системы программирования на прикладную систему необходимость в составлении алгоритма решения задачи остается.

Одним из решений проблемы является выведение на качественно новый уровень возможностей студентов для самостоятельного приобретения знаний в области алгоритмизации решения прикладных задач и получения алгоритмического подхода для применения в своей дальнейшей практике. К сожалению, необходимые традиции самообразования в стране не сложились. Но в то же время очевидны рост числа компьютеров в персональном пользовании и естественное стремление молодежи овладеть компьютерной техникой на профессиональном уровне. Это благоприятное обстоятельство позволяет ставить задачи повышения образовательного уровня посредством создания компьютерных учебных университетских курсов (компьютерных учебных лабораторий). Следует отметить и такое положительное обстоятельство, что подобный подход к самообразованию не будет являться новым либо неожиданным для наших учащихся: в коммерческой продаже пользуется спросом множество мультимедийных программ изучения курсов средней школы, в том числе и информатики. К сожалению, в этом перечне практически отсутствуют мультимедийные программы высокого качества, позволяющие приобрести знания и навыки алгоритмического подхода к решению задач. Поэтому возможное решение проблемы видится в создании эффективного электронного учебника по алгоритмизации, построенного по принципу экспертных систем и систем искусственного интеллекта.

Для того чтобы студент получил навыки алгоритмизации прикладных задач, его следует научить подходить системно и формально к такому интеллектуальному процессу, как построение алгоритма, т.е. предложить ему такие рекомендации по составлению алгоритма, которые основываются (там, где это возможно) на формальных правилах и операциях. Компьютерный учебник по алгоритмизации призван не только предоставить студенту исчерпывающий теоретический материал по данной тематике, но и с помощью удобного интерфейса путем ведения методически продуманного диалога направлять формирование алгоритма решения задачи, подсказывая в нужные моменты типовые варианты и схемы уже существующих решений. Каждый этап составления алгоритма может быть отработан отдельно на множестве небольших и доступных для понимания учебных заданий. При подборе задач для обучения алгоритмизации не всегда следует руководствоваться

правилом, что задача должна носить прикладной характер. Часть задач прикладного направления может быть успешно решена студентами только после того, как использующийся при ее решении алгоритм был проработан на абстрактном примере.

В формальном системном процессе разработки алгоритма можно выделить следующие этапы:

1. Алгоритмический анализ данных

1.1. *Выявление исходных данных.* Из условия любой абстрактной задачи, а тем более задачи прикладного характера, всегда можно определить те данные, без знания которых невозможно получить результат. Далее необходимо, если это можно сделать сразу, определить тип исходных данных – являются ли они простыми или структурированными (массивами, файлами и др.), присвоить данным имена, которыми можно будет оперировать при построении алгоритма.

1.2. *Выявление результирующих данных*. Этот процесс также основывается на внимательном изучении постановки задачи и составлении по ключевым словам и смыслу набора только тех данных, которые необходимо получить в результате решения задачи. Как и исходные данные, результирующие данные именуются и определяется их тип.

1.3. *Выявление промежуточных данных*. Этот этап невозможно выполнить полностью на начальной стадии разработки алгоритма. Если задача несложная, носит линейный характер, например, состоит из набора последовательных формул для инженерного расчета, то промежуточные данные выявляются просто. В сложном алгоритме набор промежуточных данных пополняется по мере разработки алгоритма.

2. Этап построения функциональной и структурной схемы алгоритма

2.1. *Разделение задачи на части* (подзадачи). Одной из основных особенностей алгоритмизации сложных и объемных задач является процесс выделения в задаче нескольких частей по некоторым принципам, зависящим от исходных данных, наличия циклов и т.д.

2.2. *Построение функциональной схемы алгоритма*. Когда задача разбита на подзадачи, каждая из которых имеет свои исходные данные и результаты, необходимо определить взаимосвязи между ними и уровни их иерархии. Визуально подобную разработку можно представить в виде функциональной иерархической схемы, на одном уровне которой содержатся независимые подзадачи.

2.3. *Построение укрупненной структурной схемы алгоритма*. Данный этап алгоритмического синтеза позволяет не только ука-

зать на зависимость одной подзадачи от другой, но и определить тип этой зависимости: линейный, циклический, выборный и т.д. Структурная схема алгоритма и есть, по сути дела, графическая схема алгоритма в общепринятом в информатике ее понимании. Как правило, для учебных задач упрощенная графическая схема алгоритма носит линейный, реже циклический характер.

3. Этап подбора типового алгоритма для решения подзадачи

Для реализации этого этапа следует иметь банк типовых алгоритмов, классифицированных по базовым признакам. Примерами подобных алгоритмов могут быть как простые: поиск суммы, произведения, количества, минимума, максимума в массивах и файлах данных, так и более сложные – сортировка, сложный поиск, рекурсивные алгоритмы.

4. Подбор тестовых примеров

Этот этап может выполняться в любой момент синтеза алгоритма. Процесс подбора тестов – это процесс одновременного понимания сути задачи, поэтому он может быть предложен студенту для выполнения сразу же после этапа алгоритмического анализа задачи. Подбор первичного теста позволит сформировать требования к типовым алгоритмам, затем набор тестов пополняется тестами, обрабатывающими краевые условия решения задачи.

Этапы алгоритмизации не всегда выполняются в строго определенном порядке. Студент в процессе работы над алгоритмом может возвращаться к первым этапам несколько раз, дополняя, например, список промежуточных данных уже после того, как произошел выбор типовых алгоритмов решения задачи.

На основании предложенного подхода ведется разработка гибридной экспертной системы (электронного учебника) обучения алгоритмизации для студентов технических непрофильных специальностей вузов. Обучение в системе строится на основе последовательности «Прочитай учебник – Реши с подсказкой – Реши самостоятельно» с ее отображением на следующие разделы алгоритмизации:

- алгоритмизация задач начального уровня;
- алгоритмизация задач обработки структурированных данных;
- алгоритмизация прикладных задач.

Каждый раздел, в свою очередь, содержит подразделы, которые базируются на задачах разной степени сложности. Например, для алгоритмов начального уровня подразделами являются линейные, разветвляющиеся и циклические алгоритмы. Информационной основой каждого подраздела являются задачи двух-трех уровней сложности (уровень А, уровень В, уровень С) как для самостоятельного, так и для несамостоятельного решения.

Система содержит ряд методик обучения, которые на данном этапе разработки выбираются студентом из предложенного перечня в диалоге. При дальнейшем развитии системы перед началом обучения студенту будут предложены тесты, после анализа которых система выберет методику обучения автоматизированно.

На рис. 1 приведен вид диалогового окна при выполнении первого этапа – алгоритмического анализа задачи в режиме алгоритмизации с подсказкой. Здесь производится выбор исходных, результирующих и промежуточных данных для простейшего линейного алгоритма уровня А.

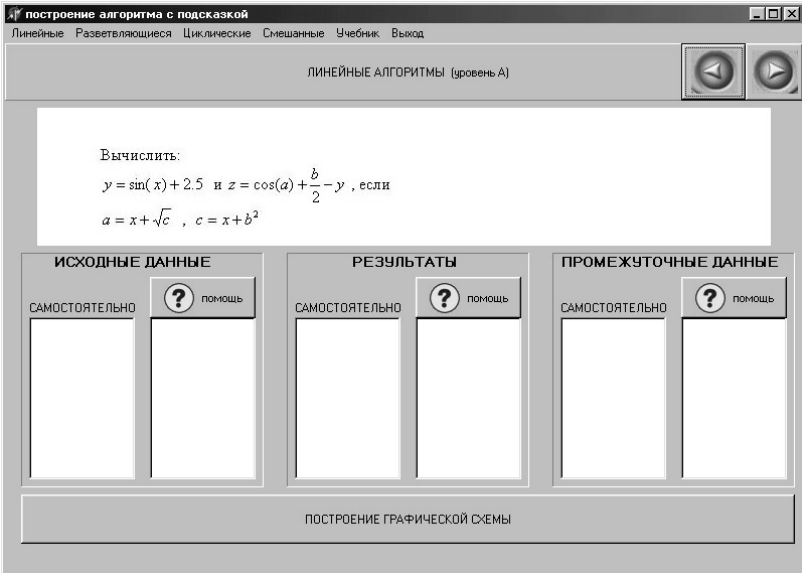

Рис. 1

На рис. 2 приведен вид диалогового окна системы при построении простейшей графической схемы алгоритма в режиме с подсказкой.

Предложенный подход к обучению алгоритмизации и система, разработанная на его основе, не могут, естественно, являться универсальным средством решения задачи. Составление алгоритма – творческий, интеллектуальный процесс, в котором ориентация на заранее выбранные схемы и модели решения не всегда приводит к положительным результатам, а иногда и усложняет исходное задание. Основная цель – научить студента

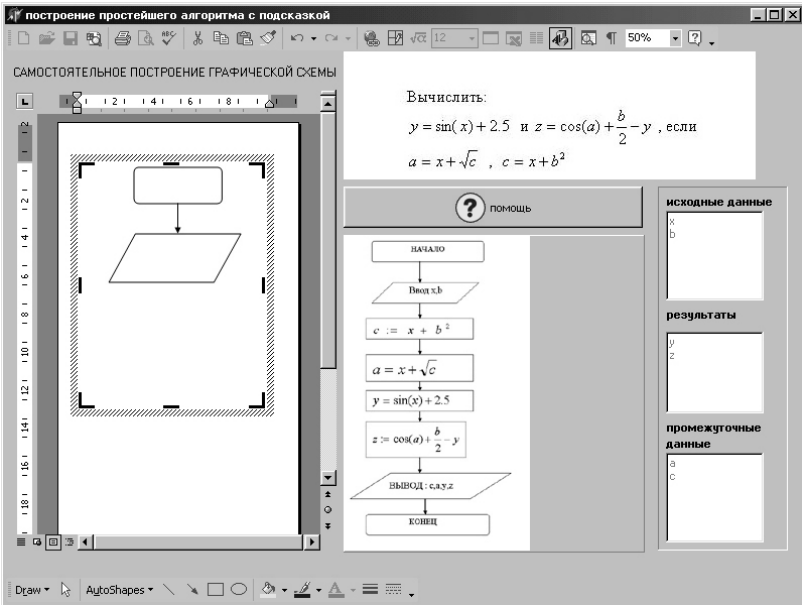

Рис. 2

системно и формально подходить к решению научных и инженерных задач, получая знания об алгоритмизации в режиме самообучения – на взгляд авторов, с помощью данной системы реально достижима.