

Таким образом, *педагогическими требованиями к осуществлению дифференцированного подхода* обучения детей в процессе их реабилитации являются: разное число часов, отводимое на изучение материала для каждого ученика; различный уровень изложения учебного материала; дифференцированный подбор разных по сложности заданий для каждого больного ребенка; дифференциация домашних и классных заданий; дополнительные и стимулирующие занятия; дифференциация заданий по степени сложности и самостоятельности ученика; разные виды педагогической помощи; эвристические беседы на уроках, занимательность и проблемность в обучении.

РЕАЛИЗАЦИЯ ПРОЦЕССА ОБУЧЕНИЯ АЛГОРИТМИЗАЦИИ С ПОМОЩЬЮ ИСПОЛНИТЕЛЕЙ ДЛЯ ШКОЛЬНОГО КУРСА ИНФОРМАТИКИ

Д.Р. Кузьмичёв

В рамках данной публикации постараемся очертить и рассмотреть с методической стороны круг нововведений, которые вносят исполнители в изучение школьного курса алгоритмизации.

В стандарте по информатике есть тема «Алгоритмы. Исполнители алгоритмов (назначение, среда, режим работы, система команд)», в ходе изучения которой в школе учащиеся должны: определять возможность применения исполнителя для решения конкретной задачи по системе его команд, строить и исполнять на компьютере алгоритм для учебного исполнителя (типа «черепахи», «робота» и т.д.).

Цель обучения алгоритмизации заключается в овладении учащимися структурной методикой построения алгоритмов. Это значит, что ученики должны уметь использовать основные управляющие структуры: следование, ветвление, цикл; разбивать задачу на подзадачи, применять метод последовательной детализации алгоритма. Дидактические средства – это разнообразные учебные исполнители: черепаха, робот*[1], чертежник, копатель и др.

Известна методическая идея А.П. Ершова: исполнители алгоритмов делятся на исполнителей, работающих «в обстановке» и исполнителей, работающих с величинами.

Использование таких исполнителей с методической точки зрения очень эффективно. Основные достоинства – понятность решаемых задач, наглядность работы исполнителя.

С большим интересом ученики 8-х и 10-х классов решают алгоритмические задачи для исполнителей. Уровни сложности таких задач можно варьировать от тривиальных до олимпиадных.

Робот

Робот – один из самых простых учебных исполнителей.

Он функционирует на прямоугольном клетчатом поле, между клетками которого могут быть стены, сквозь которые пройти невозможно. Клетки могут быть закрашены. Команды исполнителя: вверх, вниз, влево, вправо, закрасить. До недавнего времени этот исполнитель, реализованный в системе ИНТАЛ (или КуМир) был стандартом де-факто для школьных учителей. Он появлялся на уроке вместе с учебным алгоритмическим языком и в старших классах применялся для решения задач на моделирование (наряду с **Чертежником**).

Абориген

Исполнителя Абориген можно представить как некое шагающее и прыгающее устройство, способное поворачиваться налево и направо, а также менять цвет линий, которые он рисует.

Средой обитания исполнителя Абориген также является прямоугольное клеточное поле. Абориген уже (в отличие от Робота!) ориентирован в одну из четырёх сторон и изображается в виде красной стрелки. Абориген также считается непрозрачным исполнителем (в одном узле одновременно не может находиться более одного исполнителя).

Абориген может передвигаться в ближайший по направлению его ориентации узел, причём возможно передвижение с оставлением следа или без его оставления. След, который оставляет исполнитель, изображается в виде отрезка

Как видим, данный исполнитель уже приближен к «реальному» восприятию мира учеником – он имеет направление. Как показывает практика, ученик зачастую на место исполнителя ставит себя и, именно поэтому ему проще анализировать сам алгоритм, если исполнитель имеет «лицо».

Покупатель

Важной методической особенностью этого исполнителя являются команды-вопросы: Есть_Мясо, Есть_Колбаса, Есть_Хлеб, Есть_Пирожные, Есть_Молоко, Есть_Масло.

Все эти команды-вопросы планомерно подводят учеников к укоренению и осмыслению понятия «условия» в алгоритмах.

Робот*

В предложенном исполнителе запрограммированы следующие команды: «влево», «вправо», «вверх», «вниз», «СТОП», парный оператор «НЦ – КЦ» и список событий: "Слева свободно", "Сверху свободно", "Справа свободно", "Снизу свободно", "Слева стена", "Сверху стена", "Справа стена", "Снизу стена", "On Error (любая ошибка)", что является несомненным расширением стандартных возможностей исполнителей из среды КуМир и ИНТАЛ.

Как видно из набора допустимых команд, представленный исполнитель полностью обходится без условий в традиционном смысле: "нц пока", "если", "при". Обоснование такого подхода объясняется наличием **списка событий**, как более первичной и понятной ученику сущности. Таким образом, предложенный исполнитель может изучаться еще до стандартных средств алгоритмизации.

Событие «On Error (любая ошибка)» носит более общий характер чем остальные – оно дает пользователю возможность запрограммировать действия исполнителя в случае попытки выполнить любое недопустимое действие (например, если нам неизвестно, в каком месте будет стена).

Таким образом, с помощью Робот можно объединить базовые понятия объектно-ориентированного программирования и УАЯ при изучении школьного курса алгоритмизации.*

Исполнитель «Робот*» предоставляет большую свободу в организации структуры и управления роботом, качественно новый подход к программированию исполнителей такого класса (использование связки «событие» – «прерывание» для управления роботом) – все это позволяет моделировать и решать самые разнообразные учебные задачи.

Таким образом, исполнитель может использоваться как в качестве средства обучения компьютерному моделированию, так и на факультативах по информатике в качестве пропедевтики объектных языков программирования (Delphi, VBA).

В заключении хочется заметить, что количество исполнителей разработанных непосредственно учителями информатики в последние годы резко выросло. Это дает нам целый круг совершенно новых виртуальных сред, программных комплексов и, как результат, – многообразие подходов к обучению алгоритмизации.

С одной стороны появление языков высокого уровня, ориентация на них ВУЗов, заставляет меняться и школьную программу, однако основная цель исполнителей остается неизменной – обучение алгоритмизации и создание логической базы для последующего изучения языков высокого уровня.

Литература

1. Бочкин, А.И. Учебный исполнитель, управляемый событиями / А.И. Бочкин, Д.Р. Кузьмичёв // Информатизация образования. – 2008. – №1.
2. Кушниренко, А.Г. Основы информатики и вычислительной техники/ А.Г. Кушниренко, Г.В. Лебедев, Р.А. Сворень –М.: Просвещение,1993.
3. Быкадоров, Ю.А. Информатика: Учеб. пособие для 9-го кл. общеобразоват. шк./ Ю.А. Быкадоров, А.Т. Кузнецов – 2-е изд. – Мн.: Нар. асвета, 1998.– 253 с.: ил.

ПРОБЛЕМЫ СОВЕРШЕНСТВОВАНИЯ МЕТОДИКИ НАЧАЛЬНОГО ОБУЧЕНИЯ МАТЕМАТИКЕ

З.К. Левчук

Целью данного исследования является выявление возможных путей совершенствования методики начального обучения математике. Цель определила следующие задачи исследования: проанализировать научно-методическую литературу по проблеме математического образования учащихся начальных классов; определить основные направления математического развития младших школьников.

Одним из направлений обновления курса математики при выработке концепции начального математического образования А.А.Столяр назвал гуманизацию обучения путем развития младших школьников [2]. При этом под математическим развитием, как отмечается в трудах доктора педагогических наук А.В.Белюшиной, понимается «целенаправленное и методически организованное формирование совокупности взаимосвязанных основных (базовых) свойств и качеств математического мышления ребёнка и его способности к математическому познанию действительности» [1, 44].

А результатами математического развития учащихся являются: сформированные у них приёмы умственной деятельности (анализ, синтез, абстрагирование, обобщение, сравнение, классификация); умения строить индуктивные и дедуктивные рассуждения; наличие практико-