



Д. И. Павлов

Московский педагогический государственный университет, г. Москва, Россия

А. В. Каплан

*Московский педагогический государственный университет;
школа № 2009, г. Москва, Россия*

РАННЕЕ ОБУЧЕНИЕ ПРОГРАММИРОВАНИЮ: ОБЗОР ОСНОВНЫХ ЗАТРУДНЕНИЙ

Аннотация

В статье рассматриваются подходы к обучению программированию учащихся начальной школы. Акцент сделан на типовых затруднениях, которые испытывают ученики, изучая программирование, на причинах этих трудностей и возможных методах их преодоления. Представлена ретроспектива развития обучения программированию в школах России, в том числе в сравнении с международным опытом. Определено место начального курса программирования в системе образовательных результатов начального образования. На основе анализа работ А. П. Ершова, А. В. Горячева, Е. К. Хеннера, М. Gujberova, I. Kalaš, S. Grozdev, J. Wing и других специалистов установлена связь между обучением программированию и широким толкованием термина «грамотность», а также с вычислительным мышлением. Представлены результаты исследований авторов, в частности, выявленные различия в подходах к обучению программированию младших школьников, реализуемых учителями начальной школы и учителями информатики. На основе анализа научно-методической литературы выделен ряд проблем, возникающих при раннем обучении программированию, в частности, таких, как: «Самоценность компьютера в сознании ребенка», «Отсутствие связи “алгоритм — программа”», «Игра вместо учебы». В ходе практического исследования установлены причины и возможные пути преодоления этих проблем. Кроме того, в ходе исследования авторами выявлены и описаны в данной статье несколько типов проблем, которые еще требуют более глубокого изучения: «Щелчок ради щелчка», «Не читатель», «Глупый робот», «Декомпозиция», «Геометрия», «Множество одинаковых вопросов».

Ключевые слова: информатика, программирование, раннее обучение программированию, начальная школа.

DOI: 10.32517/2221-1993-2021-20-9-51-58

1. Введение

Обучение основам программирования как обязательный элемент программы подготовки школьников появилось в рамках курса «Основы информатики

и вычислительной техники» в 1985 году. Авторы курса, в частности академик А. П. Ершов, широко определяли задачи обучения программированию, выводя их далеко за рамки простой автоматизации вычислений. В своем докладе «Компьютеризация школы и математическое

Контактная информация

Павлов Дмитрий Игоревич, канд. пед. наук, ст. преподаватель, Институт математики и информатики, Московский педагогический государственный университет, г. Москва, Россия; *адрес:* 119991, Россия, г. Москва, ул. Малая Пироговская, д. 1, стр. 1; *e-mail:* fallenmonk@mail.ru

Каплан Аделя Викторовна, ассистент, факультет начального образования, Московский педагогический государственный университет, г. Москва, Россия; *адрес:* 119991, Россия, г. Москва, ул. Малая Пироговская, д. 1, стр. 1; учитель начальных классов, школа № 2009, г. Москва, Россия; *адрес:* 117042, Россия, г. Москва, ул. Адмирала Руднева, д. 16, корп. 1; *e-mail:* adel.caplan@ya.ru

D. I. Pavlov

Moscow Pedagogical State University, Moscow, Russia

A. V. Kaplan

Moscow Pedagogical State University;
School 2009, Moscow, Russia

LEARNING TO PROGRAM: AN OVERVIEW OF THE MAIN DIFFICULTIES OF THE INITIAL STAGE

Abstract

The article discusses approaches to teaching programming to primary school students. The emphasis is placed on typical difficulties that students experience while learning programming, on the reasons for these difficulties and possible methods for overcoming them. A retrospective of the development of teaching programming in Russian schools is presented, including in comparison with international experience. The place of the initial programming course in the system of educational results of primary education has been determined. Based on the analysis of the works of A. P. Ershov, A. V. Goryachev, E. K. Henner, M. Gujberova, I. Kalaš, S. Grozdev, J. Wing and other specialists, a connection has been established between teaching programming and a broad interpretation of the term "literacy", as well as computational thinking. The results of the authors' research are presented, in particular, the differences identified in the approaches to teaching programming for primary schoolchildren, implemented by primary school teachers and informatics teachers, are discussed. Based on the analysis of scientific and methodological literature, a number of problems that arise during early learning in programming are identified, in particular, such as: "Computer for the sake of the computer", "Lack of links between the algorithm and the program", "Playing instead of learning". In the course of practical research, the causes and possible ways of overcoming these problems have been established. In addition, in the course of the study, the authors identified and described in this article several types of problems that still require further study: "Click for a click", "Not a reader", "Stupid robot", "Decomposition", "Geometry", "Many the same questions".

Keywords: informatics, programming, early learning in programming, primary school.

образование», сделанном на Шестом международном конгрессе по математическому образованию (ICME-6, 1988), А. П. Ершов говорил о том, что «союз трех фундаментальных учебных дисциплин — языка, математики и информатики — образует нераздельную основу современного образования» [5].

Отмечая взгляды А. П. Ершова на информатику, в частности на ее широкий, всеобъемлющий характер, мы не забываем о самом известном его высказывании: «Программирование — вторая грамотность», которое зачастую понимается несколько упрощенно. А. В. Горячев отмечает, что «в 80-х годах программирование было единым инструментом как для разработки средств вычислительной техники, так и для применения ЭВМ в разных отраслях народного хозяйства. Но с появлением все большего количества приложений и тем самым с расширением инструментальных средств пользователя программирование частично теряет актуальность в качестве второй грамотности» [4, с. 451].

Однако, несмотря на тенденцию, отмеченную А. В. Горячевым, обучение программированию в школе не только не подвергается сокращению, но и наоборот, переживает новый рост. Причины тут не в программировании как профессиональной деятельности, а в кумулятивном эффекте обучения программированию. А. П. Ершов и его коллеги Г. А. Звенигородский и Ю. А. Первин, описывая этот эффект, говорили об операционном стиле мышления, который определяют [6]:

- умение планировать структуру действий, необходимых для достижения цели при помощи фиксированного набора средств;
- умение строить информационные модели для описания объектов и систем;
- умение организовать поиск информации, необходимой для компьютерного решения поставленной задачи;
- дисциплина и структурирование языковых средств коммуникации;
- навык своевременного обращения к компьютеру при решении задач из разных предметных областей.

С конца 1990-х — начала 2000-х годов термин «операционный стиль мышления» стал реже встречаться в научно-педагогической литературе. Соответственно, изменился подход к оценке качества обучения — вместо *количества* знаний, накопленных учеником в ходе обучения, не умаляя их роли, оценке подвергается *качественная* характеристика обучения и в первую очередь умение использовать, внедрять в практику имеющиеся знания.

Компетентностный, равно как и системно-деятельностный, подходы в обучении оказали значительное влияние на структуру и содержание школьного курса информатики в начале XXI века. Однако в настоящее время развитие курса информатики и, в частности, его содержательной линии «Алгоритмизация и программирование», происходит под влиянием новых педагогических тенденций.

Одна из этих тенденций — *развитие вычислительного стиля мышления* (англ. *computational thinking*). Термин *computational thinking* был предложен С. Пейпертом, однако, более десяти лет не привлекал к себе

внимания, так как авторская трактовка этого термина носила узкоспециализированный характер. Но в 2006 году профессор Университета Южной Калифорнии (University of Southern California) Дж. Винг опубликовала статью, в которой отмечала: «Вычислительное мышление является способом решения проблем людьми, а не попыткой уподобить человеческое мышление компьютерам. Компьютеры — скучные и нудные, а люди умны и обладают воображением. Мы, люди, делаем компьютеры эффективными. Оснащенные вычислительными устройствами, мы используем наш ум, чтобы решать проблемы, которые мы не могли решать до компьютерной эры, и создавать системы, обладающие функциональностью, ограниченной только нашим воображением» [18, с. 33]. Сегодня интерпретации понятия «вычислительное мышление» и способам формирования этого мышления посвящены сотни научных работ как в России, так и за рубежом. Так, на прошедшей в Санкт-Петербурге Международной конференции по школьной информатике ISSEP-2018 свыше трети докладов были посвящены развитию вычислительного стиля мышления дошкольников, школьников и студентов.

Исследуя феномен вычислительного стиля мышления, Е. К. Хеннер подчеркивает его преемственность и фундаментальность. В частности, он пишет: «Прежде всего, даже по формальным признакам с вычислительным стилем мышления не могут конкурировать ни “грамотность”, ни “культура”, ни “компетентность”. В исследованиях по вычислительному мышлению не раз подчеркивалось, что его не следует отождествлять ни с алгоритмическим и/или математическим мышлением, ни с компьютерной грамотностью, ни с информационной компетентностью. Алгоритмическое, логическое, системное и информационное мышление, пересекаясь с вычислительным мышлением, не исчерпывают его» [12, с. 29]. Е. К. Хеннер также отмечает, что применительно к информатике развитие вычислительного стиля мышления «отражается в смене парадигм программирования и в переходе к доминирующей в наше время объектной парадигме, включающей этапы объектно-ориентированного анализа и проектирования» [там же, с. 22]. Этот взгляд уже получил распространение в научно-педагогических кругах и является отправной точкой для разработки новых подходов к преподаванию программирования в школе [8].

Вторая тенденция, влияние которой на школьный курс информатики уже отмечено специалистами, — развитие *базовой инструментальной грамотности*. Этот термин был сформулирован участниками международного проекта «Ключевые компетентности и новая грамотность». Авторы проекта определили, что базовая инструментальная грамотность «основана на использовании современных инструментов коммуникации, опирающихся на знаковые системы, подразумевает трансформацию в современных технологических условиях привычной грамотности “читать + писать + считать” с поправкой на форматы взаимодействия и способы передачи информации, в том числе в режиме “человек — человек” и “человек — машина”» [11, с. 18].

Базовая инструментальная грамотность — сложное составное понятие, которое, по мнению авторов концепции новой грамотности, состоит из [там же]:

- читательского компонента — представляющего собой способность воспринимать и создавать информацию в различных текстовых и визуальных форматах, в том числе в цифровой среде (на естественных языках);
- математического компонента — определяющего способность применять математические инструменты, аргументацию, моделирование в повседневной жизни, в том числе в цифровой среде;
- вычислительного и алгоритмического компонента — отражающего способность воспринимать и создавать информацию на формальных языках, языках программирования.

Базовую инструментальную грамотность можно рассматривать в контексте развития упомянутых ранее идей А. П. Ершова о тройственной природе грамотности. А значит, развитие подходов к обучению программированию является одной из ключевых задач на современном этапе развития школьного курса информатики и образования в целом.

В середине 2000-х годов в школьный курс информатики было введено понятие «начальный курс информатики» с последующей его конкретизацией: «начальный (пропедевтический) курс информатики — это обучение во II—VI классах, III—VI классах или V—VI классах, в зависимости от точки входа в предмет» [1, с. 10]. Научно-методическое обоснование раннего обучения информатике приведено в работах И. Н. Антипова, А. П. Ершова, Г. А. Звенигородского, Л. Л. Босовой, С. А. Бешенкова, А. В. Горячева, Н. В. Матвеевой, А. Л. Семенова, Ю. А. Первина и целого ряда других отечественных специалистов. Их позиция созвучна мнению зарубежных экспертов [14]. В частности, профессор Comenius University (Братислава, Словакия) Иван Калаш (Ivan Kalaš) в своих работах регулярно отмечает не только важность раннего обучения информатике и программированию, но и высокий статус информатики как учебной дисциплины в программе подготовки младших школьников. В частности, профессор Калаш настаивает на обязательном, а не факультативном характере курсов информатики [17].

Таким образом, можно сделать вывод, что *проблема пропедевтического обучения программированию на уровне начального образования актуальна как для России, так и для других стран.*

2. Методы исследования проблемы пропедевтического обучения программированию на уровне начального образования

При исследовании проблемы пропедевтического обучения программированию на уровне начального образования нами были использованы как пассивные (теоретические) методы — работа с научно-методической литературой (статьями, монографиями и диссертациями по теме), так и активные (эмпирические) методы — разработка и проведение экспериментальных курсов.

При анализе научно-методической литературы было обнаружено существенное концептуальное противоречие. Одни специалисты в рамках научных конференций

и на страницах научно-методической литературы ведут споры относительно оптимального языка обучения программированию [3, 10]. В то же время другие специалисты утверждают, что выбор языка представляет собой вторичную проблему и обучение программированию, особенно на начальном этапе, — это в первую очередь работа по формированию особого стиля мышления [2].

Для изучения этого явления нами был проведен опрос учителей, в котором принял участие 301 учитель, в том числе 163 учителя информатики и 138 учителей начальных классов. Результаты опроса представлены на рисунке 1.

Участникам опроса не предлагались варианты ответа — они называли один язык по своему усмотрению и исходя из своего понимания проблематики. В результате получилось, что у учителей начальных классов в ответах отсутствуют Java, JavaScript, C++ и C#. Результаты опроса показывают серьезные расхождения в вопросе выбора первого языка при обучении программированию в понимании учителей начальных классов и учителей информатики.

Далее учителям было предложено поразмышлять над тем, что же первично — язык или формирование определенного стиля мышления, а затем они давали повторный ответ на вопрос об оптимальном языке для начала обучения программированию. Результаты повторного опроса показаны на рисунке 2.

Анализ результатов опроса показывает, что при введении альтернативного варианта ответа картина мнений учителей меняется, при этом учителя начальных классов демонстрируют большую ориентацию на формирование соответствующего стиля мышления, а наибольшую инерцию мышления демонстрируют учителя информатики — приверженцы использования языка Pascal.

В рамках настоящей работы нами был принят следующий тезис: **«На пропедевтическом этапе обучения программированию первичным является формирование основ соответствующего стиля мышления, отработка элементов которого может вестись на разных языках программирования и с помощью интерактивных обучающих сред».**

При рассмотрении вопроса о возрасте вхождения в тему нами был изучен ведущий мировой и отечественный опыт, в частности, дидактические модели обучения информатике, принятые в странах Восточной Европы [15], а также результаты экспериментов по обучению программированию в разных возрастных группах учеников [13] и учащихся разного уровня начальной подготовки [16].

Были изучены особенности, традиции и проблемы обучения программированию в школах России [9], а также перспективные подходы в этой области [7]. На основании изученных материалов был принят тезис о том, что **«точкой вхождения в курс программирования может служить начальная школа».** Однако этот тезис входит в противоречие с текущим уровнем научно-методической разработки темы. Так, к отдельным языкам и интерактивным средам для раннего обучения программированию разными авторами созданы поурочные планы и наборы дидактических материалов, однако обучение младших школьников программированию наталкивается на ряд методических проблем, для решения



Рис. 1. Результаты опроса на тему «Какой язык является оптимальным для начала обучения программированию»



Рис. 2. Результаты повторного опроса на тему «Какой язык является оптимальным для начала обучения программированию»

Таблица

Количество проведенных в ходе эксперимента курсов программирования

	Курс	Количество курсов	
		2018/2019 учебный год	2019/2020 учебный год
Уроки	Полный курс основ программирования в объеме 34 ч с использованием среды Kodu	0	1
	Полный курс основ программирования в объеме 34 ч с использованием среды Scratch	0	1
	Полный курс основ программирования в объеме 34 ч с использованием интерактивной среды Code.org	0	2
	Полный курс основ программирования в объеме 34 ч с использованием интерактивной среды Codecombat.org	1	1
	Краткий курс основ программирования в объеме 17 ч с использованием среды Kodu	2	3
	Краткий курс основ программирования в объеме 17 ч с использованием среды Scratch	1	2
	Краткий курс основ программирования в объеме 17 ч с использованием интерактивной среды Codecombat.org	2	2
Внеурочная деятельность	Полный внеурочный курс основ программирования в объеме 68 ч с использованием среды Kodu	2	2
	Полный внеурочный курс основ программирования в объеме 68 ч с использованием среды Scratch	3	2
	Полный внеурочный курс основ программирования в объеме 68 ч с использованием интерактивной среды Code.org	1	2
	Краткий внеурочный курс основ программирования в объеме 34 ч с использованием среды Kodu	1	1
	Краткий внеурочный курс основ программирования в объеме 34 ч с использованием интерактивной среды Codecombat.org	1	1

которых нет соответствующей научно-методической поддержки.

В частности, на основе анализа публикаций в научно-педагогических изданиях, а также в профессиональных сообществах в сети Интернет **можно выделить следующие группы проблем:**

- *самоценность компьютера в сознании ребенка:* для многих учеников младшего возраста использование компьютера является целью, а не средством достижения цели;
- *отсутствие связи «алгоритм — программа»:* для многих учеников написанный на бумаге алгоритм — это что-то вроде зарядки для ума или обязательной повинности перед использованием компьютера, и в представлении таких учащихся алгоритм никак не связан с написанием программы;
- *игра, а не обучение:* для многих учеников начальной школы обучение программированию представляет собой игру, что часто приводит к размыванию границ и значимости основных понятий, таких как «исполнитель», «алгоритм», «программа», «условие», «цикл».

Для проверки достоверности полученных в результате обобщения данных о проблемах раннего обучения программированию была организована эксперимен-

тальная деятельность, заключающаяся в проведении уроков и циклов внеурочных занятий по обучению программированию. Эксперимент проводился в 2018–2020 годах в школах Москвы и Московской области. Статистика эксперимента приведена в таблице.

В течение двух учебных лет усилиями шести учителей было проведено 34 курса, аудитория которых составила: по данным набора — 488 человек, а с поправкой на посещаемость и отсев обучающихся — 407 человек. В рамках курсов учителя не только проводили занятия по выбранной тематике, но и уделяли внимание диагностике выделенных проблем с целью установления их причин и взаимосвязей с различными аспектами быта, обучения и развития детей. Основным элементом диагностики являлись индивидуальные беседы с обучающимися.

3. Результаты исследования проблем пропедевтического обучения программированию на уровне начального образования

В результате проведенных исследований были получены дополнительные данные по трем выявленным ранее группам проблем.

3.1. Самоценность компьютера в сознании ребенка

Вопрос: «А компьютер сегодня будет?» — самый часто задаваемый при входе в класс. Его задают от трех до семи детей на перемене перед началом урока, и в течение всего года статистика по количеству не снижается. При этом на встречный вопрос: «Что мы будем за ним делать?» внятный ответ в первом полугодии дети не дают, а во втором полугодии ответ сводится к таким, как: «Играть» и «Ну, как на прошлом уроке».

Мы можем утверждать, что в начале обучения компьютер в сознании учеников I—IV классов является *целью*. При этом даже не целью обучения. Тут уместно провести аналогию со значением игры для дошкольников, в которой важен процесс, а не результат. Так и использование компьютера для младшего школьника интересно, в первую очередь, с точки зрения самого процесса его использования. В трех случаях из пяти ученики даже не формулируют конкретных пожеланий, им просто важно «включить и нажимать».

Наше предположение о том, что такой эффект присутствует у детей, у которых дома нет персонального компьютера или он есть, но родители сильно ограничивают ребенка в его использовании, не нашло подтверждения. Подобное поведение оказалось одинаково свойственно ученикам с различным уровнем доступа к компьютеру дома.

Для выявления причин такого отношения к использованию компьютера с учениками, родителями учеников и учителями начальных классов проводились индивиду-

альные беседы и групповые опросы, в результате которых были сформулированы причины «самоценности» компьютера в сознании ребенка:

- отсутствие правильной подготовки ребенка к использованию компьютера, в том числе отсутствие информирования его о том, что такое компьютер;
- отсутствие у детей опыта использования компьютера при решении лично значимых задач;
- отсутствие опыта использования компьютера на уроках основного цикла обучения (не на уроках информатики).

3.2. Отсутствие связи «алгоритм — программа»

В эксперименте участвовали разные группы детей. Это были и ученики с опытом бескомпьютерного изучения информатики, и прошедшие обучение по программе с использованием ПК, и вовсе не обучавшиеся ранее информатике. В программе участвовали ученики начальной школы, имеющие опыт программирования, и дети, не имеющие такого опыта. Анализ динамики преодоления рассматриваемой проблемы у учеников, имевших и не имевших ранее опыта программирования, представлен на рисунке 3.

Результаты проведенного анализа показывают, что на начальном этапе обучения проблему со связью «алгоритм — программа» чаще испытывают школьники, не имевшие предварительного опыта программирования. Однако ученики, имевшие ранее такой опыт (50 % из них пришли на курсы с уже сформированной проблемой восприятия связи «алгоритм — программа»),

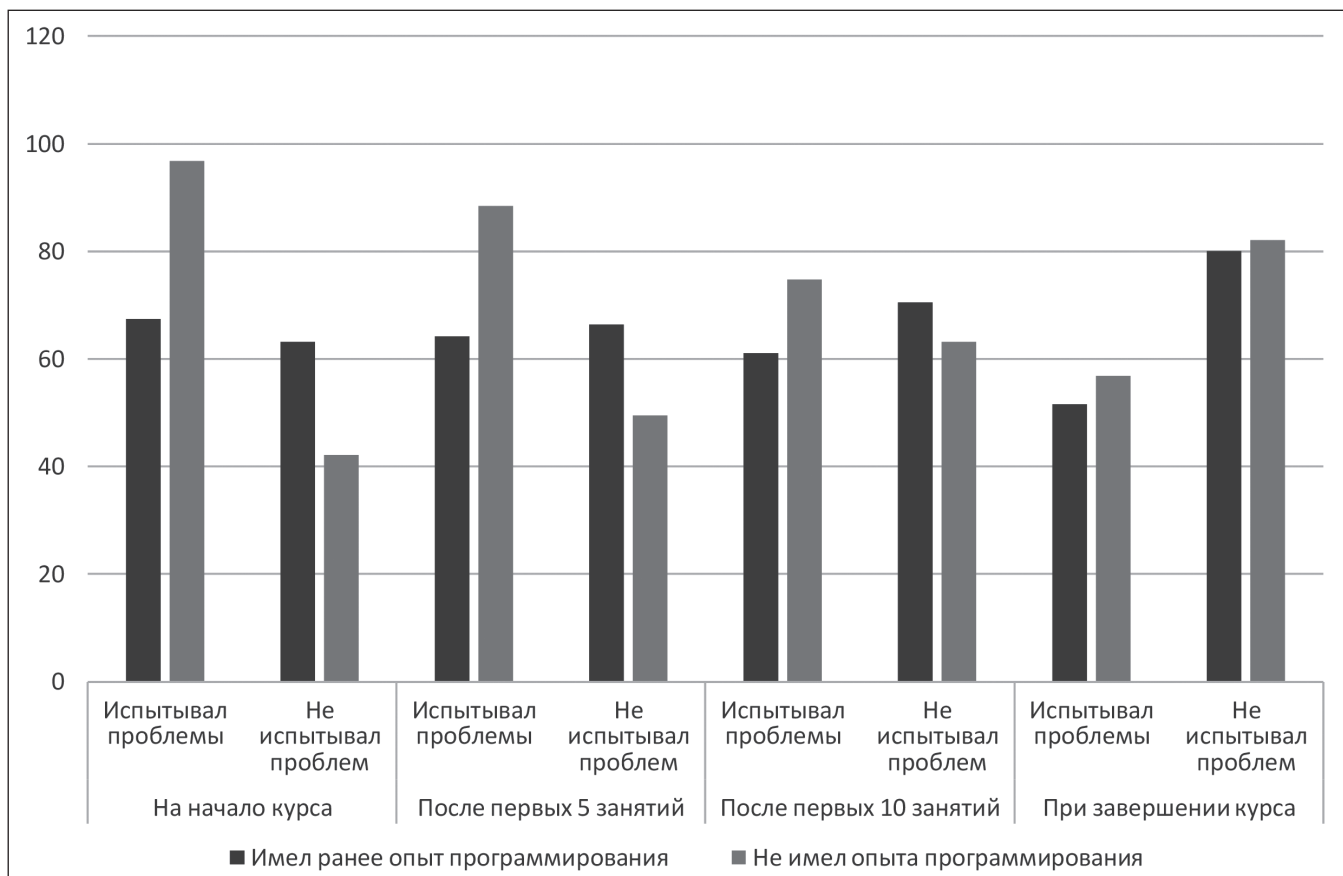


Рис. 3. Динамика преодоления трудностей связи «алгоритм — программа» у учащихся разного уровня предварительной готовности (указано количество учащихся)

демонстрируют гораздо меньшую динамику преодоления этой проблемы. На этапе завершения курса около 40 % учащихся не смогли преодолеть диагностированную проблему. Подобные показатели были зафиксированы и в группе учащихся, не имевших ранее опыта программирования, но при этом начальный показатель в этой группе был чуть меньше 70 %. Это позволяет сделать вывод, что значительная часть проблемы носит методический характер.

В целом у проблемы есть ряд предпосылок, которые удалось выявить в ходе общения с учителями и в процессе индивидуальных бесед с учащимися:

- насыщенный алгоритмической работой бескомпьютерный курс информатики;
- резкий ввод в тему программирования без предварительной теоретической подготовки;
- существенные различия между заданиями алгоритмической и программистской направленности;
- отсутствие в учебных материалах проработанных взаимосвязей «напиши алгоритм — реализуй его в среде программирования».

3.3. Игра вместо учебы

Несмотря на подъем интереса к игровым технологиям в образовании, а также возросшую популярность технологий геймификации, при раннем обучении программированию игровые формы и механики зачастую создают трудности при восприятии учащимися особенностей программирования. Так, на этапе актуализации урока на вопрос вида: «Что мы делали на предыдущем уроке?» треть учащихся, не задумываясь, отвечают: «Играли». Кроме того, желая «запустить» исполнителя и получить «одобрение» программы, обучающиеся часто не обращают внимание на наличие конкретного задания, стараясь интуитивно выполнять разные наборы действий. Установленной причиной такого восприятия можно назвать резкий контраст между избыточной академичностью большинства уроков и игровыми механиками обучения на уроках информатики или на внеурочных занятиях по программированию.

Кроме описанных выше проблем при обучении программированию младших школьников в ходе эксперимента были выявлены еще некоторые, исчерпывающий анализ причин которых не был проведен. В их числе:

- проблема «щелчков ради щелчка»;
- проблема «не читателя»;
- проблема «глупого робота»;
- проблема «декомпозиции»;
- проблема «геометрии»;
- проблема «одинаковых вопросов».

3.4. Щелчок ради щелчка

Эта проблема характерна для пиктографических сред обучения программированию типа «ПиктоМир», Kodu Game Lab и Scratch, в текстовых средах она не была выявлена.

Общее описание проблемы: ученики выбирают блок или пиктограммы с командами раньше, чем познакомятся с заданием, и зачастую не могут ответить на вопрос о том, что они делают.

На начальном этапе обучения такая проблема в разной форме была диагностирована у 70 % обучающихся. Данных по динамике преодоления затруднений и способу исправления пока недостаточно.

3.5. Не читатель

Общее описание проблемы: ученики задают вопрос: «Почему не работает?» или просят: «Помогите мне, пожалуйста, я никак не пойму, что делать». При попытке выяснить, в чем суть проблемы, обнаруживается:

- ученик и не пытался найти решение;
- ученик не прочел задание.

Особенно часто эта проблема диагностируется при выполнении заданий в «ПиктоМире», code.org, codecombat.com (до 80 % учеников на первом этапе обучения). В среде Scratch этот показатель разнится от 70 % (при введении в тему через решение задач с геометрическим и математическим содержанием) до 50 % (при введении в Scratch через анимацию).

Данных для диагностики причин такого поведения на сегодня недостаточно. Из наблюдений: учащиеся нередко просто избегают затруднения, стараясь интуитивно решить задачу, не вникая в условие. Для некоторых учеников характерна привычка к повышенной опеке и делегированию стоящих перед ними задач взрослым. Однако эти наблюдения пока не проверены.

Особой разновидностью проблемы «не читатель» является «выбор чисел» в задачах с геометрическим содержанием: при построении рисунка по образцу ученики пропускают задание, выбирая из него только числа, и методом подбора стараются установить длину, угол, количество повторений цикла, совершенно не соотнося их с задачей.

3.6. Глупый робот

Общее описание проблемы: при запуске программы, работа которой не дает желаемого результата, ученик отказывается принимать на себя роль программиста, от действий которого зависит порядок выполненных исполнителем действий, и обвиняет во всем исполнителя, компьютер, баги. Это, в свою очередь, приводит к снижению учебной мотивации и концентрации на учебной задаче.

3.7. Декомпозиция

Под декомпозицией понимается метод, когда решение одной большой задачи заменяется решением серии меньших задач, пусть и взаимосвязанных, но более простых. Применительно к пропедевтическому обучению программированию значение декомпозиции возрастает при освоении темы «Циклы».

Общее описание проблемы: обучающиеся стараются использовать цикл, не разбивая задачу на этапы. Это порождает такие команды, как «повторить 1 раз» с вложением. Иногда такие конструкции повторяются несколько раз подряд, при этом у учеников не происходит обобщения, да и сама конструкция «повторить 1 раз» не вызывает у них сомнений.

Затруднение вызывают также последовательные задачи. При выполнении рисунка квадрата с помощью цикла только 1/3 учеников II—III классов могут самостоятельно найти решение в две повторяемые четыре

паза команды. После разбора оптимального решения половина учеников не могут составить аналогичную задачу для прямоугольника.

3.8. Геометрия

Прогнозирование положения исполнителя вызывает затруднение на начальном этапе у половины учеников, но уже через 4–6 занятий эта проблема сходит на нет. Чего нельзя сказать о задачах с геометрическим содержанием. Учитывая, что на третьем–четвертом году обучения ученики не владеют должными знаниями, в частности о сумме внутренних углов многоугольника, большинство программ обучения предлагают различные методы работы, в частности методы подбора и аналогий. Однако на практике подобные методы не всегда дают стабильный результат, и ученики отказываются связывать цепочки заданий воедино и не могут воспринять работу с многоугольниками и поворотами на определенный угол.

3.9. Множество одинаковых вопросов

Нередко при выполнении задания ученик сталкивается с трудностью, задает учителю вопрос и, получив на него ответ, продолжает работу с учетом новых вводных. Однако большая часть учащихся, хотя и слышат и вопрос, и ответ своего соученика, не воспринимают его, и буквально через минуту другой ученик может задать тот же вопрос. Более того — одинаковые вопросы могут задать все ученики. А если вопросов на одном уроке возникает три–пять, то под угрозой весь ход урока. Кроме того, такие «ответы в воздух» не усваиваются обучающимися и могут повторяться из урока в урок.

4. Выводы

Результаты исследования не являются исчерпывающими. Они лишь диагностируют и систематизируют основные затруднения учащихся, однако причины каждой из проблем требуют дополнительного изучения.

Опираясь на результаты проведенной работы, можно сформулировать несколько новых направлений исследования:

- анализ существующих психолого-педагогических технологий для преодоления самооценности компьютера в сознании младшего школьника и формирования представлений о компьютере как об инструменте;
- анализ существующих педагогических технологий, форм и приемов работы и их адаптация к решению проблемы разрыва взаимосвязи между алгоритмом и программой;
- поиск баланса между игрой и обучением программированию с использованием игровых технологий;
- для проблем «Щелчок ради щелчка», «Не читатель», «Глупый робот», «Декомпозиция», «Геометрия» и «Множество одинаковых вопросов» необходимо проведение дополнительных исследований с целью выявления причин и закономерностей проявления этих проблем при обучении программированию младших школьников.

Список использованных источников

1. Бешенков С. А., Ракитина Е. А., Матвеева Н. В., Милохина Л. В. Непрерывный курс информатики. М.: БИНОМ. Лаборатория знаний, 2008. 216 с.
2. Гладков В. П. Методы обучения программированию // Вестник ПНИПУ. Электротехника, информационные технологии, системы управления. 2009. № 3. С. 94–101.
3. Городняя Л. В., Боярских А. С. О языке начального обучения параллельному программированию // Вестник НГУ. Серия: Информационные технологии. 2012. № 2. С. 85–100.
4. Горячев А. В. О целесообразности модульной организации курса информатики в основной и старшей школе // Сборник материалов международной научной конференции «Информатизация непрерывного образования — 2018» (г. Москва, 14–17 октября 2018 года). Т. 2. М., 2018. С. 450–453.
5. Ершов А. П. Компьютеризация школы и математическое образование. Пленарный доклад // Шестой международный конгресс по математическому образованию (ICME-6). http://ershov-arc.iis.nsk.su/russian/articles_pdf/23.pdf
6. Ершов А. П., Звенигородский Г. А., Первин Ю. А. Школьная информатика (концепция, состояние, перспективы). Препринт ВЦ СО АН СССР. № 152. Новосибирск, 1979 // Архив академика А. П. Ершова. <http://ershov.iis.nsk.su/ru/node/805749>
7. Каплан А. В., Павлов Д. И. Разработка методических подходов к реализации пропедевтического курса информатики в начальной школе средствами Kodu Game Lab // Информатика и образование. 2019. № 8. С. 14–23. DOI: 10.32517/0234-0453-2019-34-8-14-23
8. Павлов Д. И., Бутарев К. В., Балашова Е. В. О перспективах использования технологий геймификации при раннем обучении объектно-ориентированному программированию // Современные информационные технологии и ИТ-образование. 2018. Т. 14. № 4. С. 977–985.
9. Первин Ю. А. Проблемы раннего обучения информатике в российской школе // Вопросы образования. 2005. № 3. С. 166–182.
10. Родыгин Е. Ф. Методические рекомендации обучения программированию в школе // Вестник Марийского государственного университета. 2011. № 7. С. 20–22.
11. Фрумин И. Д., Добрякова М. С., Баранников К. А., Реморенко И. М. Универсальные компетентности и новая грамотность: чему учить сегодня для успеха завтра. Предварительные выводы международного доклада о тенденциях трансформации школьного образования. М.: НИУ ВШЭ, 2018. 28 с. (Современная аналитика образования. 2018. № 2 (19).) https://ioe.hse.ru/data/2018/07/12/1151646087/2_19.pdf
12. Хеннер Е. К. Вычислительное мышление // Образование и наука. 2016. № 2. С. 18–31.
13. Blaho A., Kalaš I., Matusova M. Experimental curriculum of informatics for 11 year old children // IFIP World Conference on Computers in Education. Springer, Boston, MA, 1995. P. 829–841.
14. Coding, programming and the changing curriculum for computing in schools. UNESCO, 2019. 45 p. <https://ru.scribd.com/document/454660001/OCCE-2018-TC3-UNESCO-meeting-040219-CS-coding-pdf>
15. Grozdev S., Terzieva T. A didactic model for developmental training in computer science // Journal of Modern Education Review. May 2015. Vol. 5. No. 5. P. 470–480.
16. Gijberova M., Kalaš I. Designing productive gradations of tasks in primary programming education // Proceedings of the 8th Workshop in Primary and Secondary Computing Education. ACM, 2013. P. 108–117.
17. Kabátová M., Kalaš I., Tomcsányiová M. Programming in Slovak primary schools // Olympiads in Informatics. 2016. Vol. 10. P. 125–159.
18. Wing J. Computational Thinking // Communications of the ACM. 2006. Vol. 49. No. 3. P. 33–35.