Онлайн-доступ к журналу: http://journal.iro38.ru

УДК 372.8

Роль автоматизированных тестирующих систем в процессе формирования навыков алгоритмического мышления

О. В. Зубков, Н. Л. Семичева

Иркутский государственный университет, г. Иркутск

Аннотация.

Введение. *В статье рассмотрены вопросы формирования у школьников навыков алгоритмического мышления на уроках информатики и факультативных занятиях по программированию. Проведён анализ современных методов обучения и автоматизированных тестирующих систем (комплекс программных средств, позволяющих в автоматическом режиме проводить тестирование решений пользователей), которые позволяют как развить базовые навыки, так и вывести учащихся на высшую ступень мастерства в области программирования.*

Материалы и методы. *Исследуется проблема естественного разделения учащихся во время этих занятий на подгруппы по уровню подготовки и особенностям мышления. Проанализирован подход к её решению с использованием автоматизированных тестирующих систем. Проведён общий обзор и классификация наиболее распространённых ресурсов с автоматической проверкой программ.*

Результаты исследования. *В данном разделе показано, какие результаты были получены при использовании представленной методики, и даны общие рекомендации по её распространению.*

Заключение. *Представленная методика может быть использована на уроках информатики как для общего развития алгоритмического мышления школьников, так и для базовой подготовки специалистов, деятельность которых будет существенно определять направления развития информационной сферы в будущем.*

Ключевые слова: *алгоритмическое мышление, программирование, работа в малых группах, автоматизированная тестирующая система.*

Для цитирования: *Зубков О. В., Семичева Н. Л.* Роль автоматизированных тестирующих систем в процессе формирования навыков алгоритмического мышления // Педагогический ИМИДЖ. 2019. № 4 (45). С. 550–565.
DOI: 10.32343/2409-5052-2019-13-4-550-565

Дата поступления статьи в редакцию: 11 апреля 2019 г.

Введение

Информационная сфера является в настоящий момент наиболее быстро развивающейся и одной из самых значимых в современном мире. Это подтверждается и поддержкой развития информационных технологий в Российской Федерации, осуществляемой на государственном уровне, о чём свидетельствуют разработанные Правительством Российской Федерации программа «Информационное общество (2011–2020 годы)» и «Стратегия развития отрасли информационных технологий в Российской Федерации на 2014–2020 годы и на перспективу до 2025 года», что говорит о наличии финансирования и плановом увеличении числа рабочих мест в данной области в нашей стране.

Как следствие, имеется всё более возрастающий спрос на специалистов, владеющих на профессиональном уровне необходимыми компетенциями и навыками работы в современных системах программирования.

В настоящее время недостаточно быть просто программистом-исполнителем: специалистов подобного рода в больших количествах готовят в таких густонаселённых странах, как Индия и Китай. Помимо собственно владения методами работы в системах и средах программирования постоянно возникает необходимость в совершенствовании и улучшении самих инструментов разработки, отладке и внедрении новых программных продуктов. Это приводит компании, сферой деятельности которых является разработка и поддержка программного обеспечения, к необходимости поиска специалистов с широким кругозором, творческим мышлением и умением «увидеть» и спроектировать новые, более эффективные и удобные инструменты и программы.

Для того, чтобы получить специалиста такого уровня, необходимо сформировать у него определённый набор компетенций, в основе многих из них лежит алгоритмическое мышление, которое необходимо начинать развивать ещё в школе. Рассмотрим, как автоматизированные тестирующие системы могут помочь в формировании навыков алгоритмического мышления.

Обзор литературы

Рассмотрим, что понимается под термином «Алгоритмическое мышление» в современном педагогическом сообществе. «Алгоритмическое мышление представляет собой специфический стиль мышления, предполагающий наличие мыслительных схем, которые способствуют видению проблемы в целом, решению задач крупными блоками с последующей детализацией и осознанному закреплению результатов решения. Также алгоритмическое мышление представляет собой набор определённых последовательностей действий, которые вместе с логическим и образным мышлением увеличивают интеллектуальные способности человека и его творческий потенциал» [8]. Алгоритмическое мышление – это «совокупность мыслительных действий и приёмов, нацеленных на решение задач, в результате которых создаётся алгоритм… Такой способ мышления отличается формальностью, логичностью, способностью облечь любую абстрактную идею в последовательную инструкцию, пошаговое выполнение которой воплощает эту идею в жизнь. Именно такое мышление способствует успешному изучению программирования» [15]. В приведённых определениях не только даётся понятие алгоритмического мышления, в последнем дополнительно показывается, какие действия необходимо предпринимать для его развития, а первое – определяет цель такого развития для всех учеников.

Исследования в области алгоритмического мышления и его развития проводятся, естественно, не первый год. Например, А. И. Гезейкина отмечала такую проблему: несмотря на то, что программирование включено в государственный образовательный стандарт, только 10–15 % от общего числа студентов, обучающихся по направлению «Информатика», являются успешными в данной области [3]. В этой же работе содержится следующее утверждение: «…программирование является специфическим видом человеческой деятельности, для успешной реализации которой необходимо не только применение приобретённых в процессе обучения знаний и умений, но требуется и наличие определённого стиля мышления», в частности алгоритмического, которое необходимо тренировать специальными методами на операционном, алгоритмическом и объектном уровнях.

В работах, посвящённых исследованию данной темы, приводятся различные методики формирования алгоритмического мышления на разных ступенях развития. Особый интерес представляют выводы Н. Н. Еремеевой [7], исследовавшей данный вопрос в начальной школе. Она показывает, что для достижения наилучшего результата необходимо делить детей на малые группы, чтобы они, выдвигая идеи решения задачи, высказывали их не учителю, а одноклассникам, не боясь ошибиться. В начальной школе многие показывают, как можно развивать алгоритмическое мышление на уроках математики [1; 2; 7]. Следует отметить, что вопросы формирования алгоритмического мышления в младших и средних классах школы имеют большую популярность и за рубежом. Различные методики, связанные с игровыми и цифровыми инструментами [19], визуальными и анимационными образами [15], а также методики оценивания успешности усвоения алгоритмических навыков [17] регулярно изучаются и обсуждаются в педагогических сообществах разных стран.

При рассмотрении вопросов формирования алгоритмического мышления в

средней школе зачастую рассматривают частные вопросы: использование рекурсивных алгоритмов при формировании алгоритмического мышления [12], использование наглядных задач при изучении базовых алгоритмических конструкций [10]. Л. Г. Лучко, наоборот, старается подойти к данному вопросу комплексно и предлагает некоторую стратегию формирования алгоритмической культуры в процессе обучения базовому курсу информатики [13].

В данной работе обсуждаются сложности в формировании алгоритмического мышления на уроках информатики и предложен кардинально иной метод его формирования, основанный на применении специализированных программных продуктов и web-сервисах с использованием баз знаний.

Материалы и методы

Следует признать, что на уроках информатики алгоритмическое мышление можно сформировать только на базовом уровне, а этого недостаточно для описанных выше целей. Здесь необходим метапредметный подход с преобладающей долей математики [14]. О глубокой связи математического и алгоритмического мышления также говорит известный американский математик, основоположник и идеолог современного программирования Дональд Кнут, который в своей статье [18] пытается найти ответ на важный вопрос: какова реальная роль понятия алгоритма в математических науках?

Естественно, уроки информатики играют важную роль в формировании этого навыка в силу своей специфики. Для развития алгоритмического мышления необходимо решать большое количество задач, связанных с разработкой и реализацией алгоритмов. А это находится в самой прямой связи с написанием программ на одном из языков программирования на уроках информатики и факультативных занятиях.

Отметим, что при работе с небольшой группой в 5–6 человек, не говоря уже о полном классе, неизбежно возникает проблема резкого разделения, обусловленная различиями в первоначальном владении материалом, скоростью восприятия информации, а также особенностями мышления учеников.

Можно выявить три основных умения, по которым происходит это разделение: умение быстро разработать алгоритм решения задачи, наличие ошибок в первоначальном программном коде, умение найти и исправить синтаксические и алгоритмические ошибки.

Эти критерии делят аудиторию на несколько кардинально различающихся групп:

1) тех, кто быстро пишет работающие программы;
2) тех, кто может придумать алгоритм, но не может его правильно написать;
3) тех, кому тяжело даётся разработка самого алгоритма.

При этом ошибки у каждого свои, и провести общее для всех занятие, на котором все ученики получат ответы на все свои индивидуальные вопросы, практически невозможно. Это разделение никак не может быть поставлено в вину ни педагогу, ни ученикам, но с течением времени лишь усугубляется. По этой причине при правильном подходе занятие с большой группой превращается в серию одновременных индивидуальных занятий или занятий в малых группах, когда учитель ставит перед каждой подгруппой учеников посильную задачу, а затем помогает найти и исправить ошибки как в алгоритме, так и в

реализующей его программе. Только таким образом можно достичь общей вовлечённости в учебную деятельность.

Если подходить к процессу развития алгоритмического мышления с таких позиций, то учитель должен уметь:

- эффективно оценить общий уровень подготовленности ученика и подобрать для него соответствующую его уровню задачу;
- наглядно и доступно объяснить алгоритм решения этой задачи, если ученик не справился с заданием, разработать его самостоятельно;
- быстро найти и исправить незначительную ошибку технического плана в «почти работающей» программе ученика;
- быстро найти пример входных данных, при которых программа будет давать неправильный ответ, что позволит ученику обнаружить и исправить ошибку самостоятельно;
- объяснить, почему разработанный учеником алгоритм в принципе непригоден или малоэффективен и предложить пути для его улучшения;
- дать окончательную экспертную оценку программы, написанной учеником.

Если учесть, что учителю нужно вести занятие одновременно в нескольких подгруппах, работающих в разном темпе, весьма полезной представляется возможность автоматизировать некоторые из вышеперечисленных функций.

Практика проведения таких занятий показала, что эффективную помощь в обучении навыкам алгоритмизации и программирования могут оказать автоматизированные тестирующие системы и ресурсы.

«Автоматизированная тестирующая система (АТС) – это комплекс программных средств, которые позволяют в автоматическом режиме проводить тестирование решений пользователей… Говоря о задачах по программированию, важно отметить, что они кардинально отличаются от всех остальных. … решением задачи в программировании является программа, которая решает целый класс однотипных задач, отличающихся входными данными» [4]. Иными словами, принципиальное отличие здесь заключается в том, что тестируется не человек, а написанная им программа.

Как правило, основным ядром таких систем являются архив задач и, собственно, проверяющая система. Общий принцип работы в этих системах заключается в следующем:

- в архиве задач выбирается задача необходимой сложности на заданную тему;
- ученик читает условие задачи, разбирает один или несколько примеров, после чего пишет программу, решающую, с его точки зрения, эту задачу на языке программирования, доступном в системе;
- написанная программа должна иметь строго указанный в условии формат чтения/вывода данных, что обусловлено формальностью автоматической проверки;
- после написания ученик самостоятельно высылает программу на проверку в систему при помощи встроенного web-интерфейса;
- при проверке программа проходит компиляцию и запускается на множестве неизвестных ученику тестов;
- результаты проверки сообщаются ученику либо в виде таблицы, в кото-

рой указаны результаты тестирования, либо в виде общего вердикта «принято/не принято»;

– в случае вердикта «не принято» ученик должен понять его причину, внести необходимые исправления и выслать программу на повторную проверку.

При такой форме работы ученик может отработать овладение такими важными на этапах программирования умениями, как:

– понимание и формализация задачи;
– разработка и реализация эффективного алгоритма;
– тестирование и отладка программы;
– сдача готового продукта.

Причём автоматизированные тестирующие системы позволяют значительно увеличить эффективность работы с первыми двумя описанными выше группами учеников. Покажем, за счёт чего это достигается.

Ранее мы обозначили, какие задачи должен решать учитель при обучении алгоритмическому мышлению. В таблице 1 проведём параллель между этими задачами, функциями АТС и механизмами их использования в учебном процессе для повышения эффективности работы на уроках.

Таблица 1

Повышение эффективности работы на уроках и факультативах по программированию с использованием автоматизированных тестирующих систем

Table 1

Improvement of the efficiency of work in programming classes and electives using automated testing systems

Задачи учителя в рамках урока по программированию	**Функции АТС**	**Использования АТС на уроке**
Эффективно оценить общий уровень подготовленности ученика и подобрать для него соответствующую его уровню задачу	Задачи структурированы по темам и сложности	Ускоряется процесс подбора задач учителем
Наглядно и доступно объяснить алгоритм решения этой задачи, если у ученика не получается разработать его самостоятельно	В большинстве случаев задачи сопровождаются комментариями, обсуждениями, разборами	Учитель помогает ученику в разработке алгоритма только в том случае, если он не смог разработать алгоритм сам и ему не помогли дополнительные материалы задачи
Быстро найти и исправить незначительную ошибку технического плана в «почти работающей» программе ученика	Программа проходит компиляцию в системе	При компиляции указывается позиция и тип ошибки, что в большинстве случаев позволяет ученику выявить её самостоятельно

Быстро найти пример входных данных, при которых программа будет давать неправильный ответ, что позволит ученику обнаружить и исправить ошибку самостоятельно	При проверке программа запускается на множестве неизвестных ученику тестов	Некоторые системы показывают тесты, на которых программа работает неправильно, что позволяет ученику найти ошибку в своём алгоритме
Дать окончательную экспертную оценку программы, написанной учеником		Оценку правильности программы система выносит автоматически на основе тестирования

В результате того, что рутинные функции перекладываются «на плечи» АТС, на занятии учитель может сосредоточиться на самом главном – диалоге с учениками, обсуждении их алгоритмов и программ, поиске ошибок и методов их отладки.

Для подтверждения полученных выводов представим результаты одного из педагогических экспериментов, проведённых в рамках описанного исследования. Эксперимент заключался в том, чтобы научить две группы испытуемых в рамках подготовки к единому государственному экзамену по информатике и ИКТ решать задания 24 и 25.

В кодификаторе [9] перечислены алгоритмические задачи, которые должен научиться решать ученик, чтобы успешно справиться с заданиями ЕГЭ. Из них к заданиям 24 и 25 относятся следующие:

– записывать натуральное число в позиционной системе с основанием, меньшим или равным 10, обрабатывать и преобразовывать такую запись числа;

– находить суммы, произведения элементов данной конечной числовой последовательности (или массива);

– использовать цикл для решения простых переборных задач (поиск наименьшего простого делителя данного натурального числа, проверка числа на простоту и т. д.);

– находить минимальное (максимальное) значение в данном массиве и количества элементов, равных ему, за однократный просмотр массива;

– выполнять операции с элементами массива, отобранными по некоторому условию (например, находить минимальный чётный элемент в массиве, находить количество и сумму всех чётных элементов в массиве);

– заполнять элементы одномерного и двумерного массивов по заданным правилам.

Данная тема выбрана не случайно. Если смотреть статистику выполнения заданий участниками ЕГЭ [11], тема «Алгоритмизация и программирование» является сложной для всех уровней подготовки.

Проблема при проведении эксперимента заключалась в том, что:

– участники эксперимента были из разных школ, поэтому уровень подготовки также был очень разный: от нулевого, то есть ученики изначально не знали ни одной команды программирования, до продвинутого, когда ученики владели всеми требуемыми для решения задач навыками;

– на прохождение данной темы в рамках курса было выделено всего 6 аудиторных часов, причём на отработку алгоритмов – только 4 (что соответствует

ограниченности временных ресурсов при прохождении данной темы в школе);

– участники эксперимента в школах изучали разные языки программирования (в одной школе, например, Pascal, во второй – C++, в третьей – Pyton и так далее).

Для участников эксперимента провели вводное тестирование, состоящее из 10 задач разного уровня. По результатам вводного тестирования и анализа статистики выполнения домашнего задания по предыдущим темам участники эксперимента были разделены на две группы, соответствующие по уровню начальной подготовки по данной теме и качеству выполнения домашнего задания. В таблице 2 приведены результаты тестирования для двух групп (показано, сколько человек из каждой группы с каким количеством задач справились).

Таблица 2

Результаты вводного тестирования

Table 2

Results of introductory testing

Кол-во решённых задач	Группа 1	Группа 2	Кол-во решённых задач	Группа 1	Группа 2
0	4	3	6	2	2
1	2	2	7	0	1
2	1	2	8	1	1
3	2	2	9	2	1
4	3	2	10	0	1
5	1	0			

Так как у большинства участников эксперимента навыки программирования были очень низкими, задачи для тренировки нужных навыков были разбиты на группы:

– нахождение суммы, количества, произведения элементов во вводимой последовательности;

– нахождение минимального и максимального элемента в последовательности;

– решение перечисленных выше задач с дополнительным условием (например, нахождение суммы чётных элементов последовательности);

– решение перечисленных выше задач с несколькими условиями (например, нахождение суммы чётных отрицательных элементов последовательности);

– объявление, заполнение массива элементов, изменение и вывод конкретных элементов в массиве и всех элементов массива;

– решение первых четырёх задач не для последовательности элементов, а для массива чисел;

– выделение цифр числа, записанного в десятичной системе счисления, составление числа в десятичной системе счисления с использованием заданного набора цифр;

– работа с цифрами десятичного числа, изменение десятичного числа, решение первых четырёх задач для последовательности цифр заданного десятичного числа.

Приступая к очередной группе заданий, ученик решал тестовую задачу, проверяющую сформированность его навыков в решении задач данной группы. Если справлялся с задачей, мог перейти к следующей группе. В противном случае необходимо было выполнить все задачи данной группы, учитывающие различные нюансы и ошибки, которые встречаются при решении соответствующих задач (ошибка инициализации переменной, хранящей искомое значение, выход за пределы массива, ошибки в сложном условии, отсутствие объявления цикла, ошибки вывода данных, непонимание разницы между значением и индексом элемента массива и др.).

Набор тренировочных заданий для обеих групп был одинаковым. Все задания были введены в АТС «Contester». Участники обеих групп использовали ту среду программирования, с которой они работали в школе, участники группы 1 дополнительно использовали АТС «Contester».

Гипотеза эксперимента заключалась в том, что использование АТС позволит сформировать нужные навыки у большинства участников группы несмотря на разницу в начальной подготовке группы и небольшое количество аудиторных часов за счёт того, что задачи проверяются автоматически и ученики видят, где они допускают ошибки (если они есть): в компиляции или в алгоритме.

В ходе работы над данной темой ученики с хорошим уровнем подготовки отрабатывали свои навыки на контрольных заданиях, участники группы 1 могли также работать с задачами, содержащимися в АТС «Contester» (список задач предоставлялся), остальные трудились над решением задач нужной группы в рамках аудиторных занятий и в качестве самоподготовки дома. На аудиторных занятиях участники группы задавали вопросы по тем заданиям, с которыми они не справились. Заметим, что участники группы 2 задавали больше вопросов, учитель зачастую не успевал вовремя на них отвечать, в результате чего ученик не получал ответ или вынужден был сидеть без дела в ожидании ответа.

После окончания отведённого на изучение темы времени было проведено повторное тестирование, результаты которого приведены в таблице 3.

Таблица 3

Результаты вводного тестирования

Table 3

Results of introductory testing

Кол-во решённых задач	Группа 1	Группа 2	Кол-во решённых задач	Группа 1	Группа 2
0	0	0	6	4	3
1	0	0	7	3	2
2	0	2	8	2	1
3	1	1	9	3	2
4	1	2	10	2	1
5	2	3			

Проведённый эксперимент подтверждает, что использование АТС повышает эффективность работы при развитии навыков алгоритмизации и программирования.

Получается, что специализированные сервисы можно использовать на уроках информатики, они необходимы на факультативах по программированию для повышения эффективности обучения. Но прежде чем начать соответствующую практику, необходимо разобраться, какие особенности есть у разных сервисов, что лучше использовать на начальном уровне, а что на продвинутом. Для этого приведём краткий обзор и характеристику доступных ресурсов с автоматической проверкой.

Надо отметить, что на упомянутых ресурсах доступны все используемые на сегодняшний день в учебном процессе языки программирования: Pascal, C++, Java, Python, Basic, C# и множество других, – а значит, они могут быть встроены в обучение в любой образовательной организации.

АТС можно условно разделить на локально установленные, доступные только по локальной сети учебного заведения и общедоступные, размещённые на известных сайтах.

К достоинствам локально установленных проверяющих систем можно отнести полный контроль над всеми процессами, происходящими при работе таких систем. Оператор (учитель) может включить/отключить систему в нужный момент, «видит» все посланные учениками на проверку программы, имеет возможность посмотреть, на каком тесте и по какой причине не работает та или иная проверяемая программа. Он может, при необходимости, добавить или удалить тесты, разработать и добавить нужную на данный момент задачу. Трудности работы с такими системами тоже очевидны: для успешной установки, настройки и поддержки системы в рабочем состоянии оператор должен обладать соответствующими навыками: добавляемые задачи требуют кропотливого этапа разработки, настройки и отладки.

В качестве примера такой системы можно привести Contester (доступна и в форме интернет-ресурса). Система распространяется свободно и бесплатно, полностью открыта для дополнения и модернизации контента, может быть рекомендована для начинающих администраторов. [6]

К промежуточным системам (между локальными и общедоступными) относится система администрирования турниров ejudge. На её основе регулярно проводятся олимпиады в Иркутске и Иркутской области, а также занятия и кружки по программированию. Опыт пользования этой системой показал, что она комфортна для проведения занятий, позволяет быстро сделать нужную настройку для проверки задачи и проведения турнира (возможно, даже в рамках одного урока). Система в какой-то мере может считаться общедоступной, так как внедрена на сайтах многих учебных заведений, в том числе и на сервере Иркутского государственного университета по адресу: olymp.isu.ru. В работе «Обучение программированию с использованием системы Ejudge» [5] авторы делятся опытом использования системы ejudge в Томском государственном педагогическом университе (ТГПУ).

Переходя к обзору основных ресурсов, доступных в сети «Интернет», следует заметить, что достоинства и недостатки систем меняются местами. Достоинства очевидны: системы полностью настроены и круглосуточно используются, архивы задач имеют большие размеры (не менее тысячи задач) и постоянно пополняются, они структурированы и часто содержат дополнительные ссылки обучающего характера, увеличивается количество доступных

языков программирования, для использования ресурса нет необходимости изучать его структуру, особенности настройки. С другой стороны, преподаватель не может контролировать ход проверки программы ученика, не знает тестов и, соответственно, должен проявить большую компетентность при ответе на вопрос: «Почему моя программа не работает?» Для этого, как правило, необходимо предварительно решить эту задачу, возможно, разными способами. Благодаря опыту практического использования этих ресурсов, можно убедительно рекомендовать их для проведения как основных уроков по информатике, так и факультативов по программированию.

Кратко перечислим наиболее популярные ресурсы, относящиеся к данному типу, и, главное, отметим способы их использования.

Acmp.ru – сайт «Школа программиста», г. Красноярск. Позиционируется как система для обучения школьников, имеет раздел «Курсы», где структурированы задачи на самые простые темы по освоению языков программирования (операторы присваивания, условные операторы, операторы цикла и т. д.). Может быть рекомендован для начинающих учеников и учителей.

Acm.timus.ru – сайт сотрудников и студентов Уральского федерального университета имени первого Президента России Б. Н. Ельцина (УрФУ) содержит более сложные по сравнению с acmp.ru темы, поэтому рекомендуется для мотивированных школьников, содержит в составе архива задачи школьных олимпиад Уральского региона.

Informatics.msk.ru – московский сайт «Дистанционная подготовка по информатике», имеет форму интерактивного учебного пособия по программированию, алгоритмам и языкам, рекомендован ведущими педагогами, в том числе и К. Ю. Поляковым, имеет большой и подробно структурированный по темам архив задач.

Codeforces.ru – портал для программистов, разработан и поддерживается силами сотрудников и студентов Саратовского национального исследовательского государственного университета им. Н. Г. Чернышевского. По сравнению с ранее указанными ресурсами имеет ряд принципиальных отличий, полному описанию которых можно посвятить отдельную статью. Следует сказать, что работа этого ресурса строится не вокруг архива задач (хотя он есть, и, пожалуй, является самым большим), а вокруг регулярных онлайн-турниров, что позволяет характеризовать его именно как сайт для олимпиадного программирования. С другой стороны, открытость тестов и решений других участников для общего просмотра делают его намного более эффективным инструментом для обучения и самообучения, а система блогов и форумов, в которых авторы делятся разборами задач и ссылками на другие необходимые ресурсы, позволяет позиционировать этот ресурс как самый полезный для мотивированных учеников.

Результаты исследования

Указанные ресурсы используются при проведении уроков информатики и факультативных занятий в ряде школ Иркутска и Иркутской области, среди воспитанников которых много победителей и призёров муниципальных, региональных и всероссийских олимпиад, что подтверждает эффективность использования данных ресурсов.

За период с 2014 года ученики, для подготовки которых использовались ав-

томатизированные тестирующие системы:

– 12 раз становились победителями региональных этапов Всероссийской олимпиады школьников по информатике и ИКТ;

– 4 раза призёрами заключительных этапов Всероссийской олимпиады школьников по информатике и ИКТ;

– 3 раза призёрами Всероссийских командных олимпиад школьников по программированию;

– два участника были приглашены на всероссийский форум профессиональной ориентации «ПроеКТОриЯ 2018» (направление «Информационные технологии») и присутствовали на всероссийском открытом уроке профессиональной навигации с участием Президента РФ В. В. Путина;

– один выпускник, на данный момент студент Московского физико-технического института (национальный исследовательский университет) (МФТИ), в составе команды этого вуза занял 10 место (из 135 вузов мира) в финале студенческого командного чемпионата мира по программированию ACM ICPC и получил бронзовую медаль этого престижного соревнования.

Чтобы достичь такого результата, необходимо понимать, когда, как и какие АТС полезно использовать. Приведём основные выводы исследования, для этого представим в списке задачи обучения, наиболее подходящие для их решения АТС, а также обоснование их использования.

1. Для основы программирования наиболее подходящими являются contester и acmp.ru вследствие наличия большого числа простых задач, направленных на изучение простых языковых конструкций и структурированных по темам (acmp.ru), а также возможности неограниченного добавления своих задач на интересующую тему (contester).

2. Для решения задач, развивающих навыки разработки алгоритмов, – acmp.ru, acm.timus.ru и codeforces.ru, так как структурирование архива задач по темам и сложности позволяет для любого известного алгоритма быстро подобрать сет задач необходимого уровня сложности.

3. Для решения задач, затрагивающих тонкости языка программирования, – informatics.msk.ru, codeforces.ru – из-за наличия большого числа сопровождающих статей и комментариев по интересующим разделам программирования (informatics.msk.ru) и профессиональных блогов с обсуждением тонкостей написания программного кода, а также открытость тестов и доступность просмотра решений других участников (codeforces.ru).

4. Для олимпиадного программирования – acmp.ru, acm.timus.ru, codeforces.ru и informatics.msk.ru – в связи с возможностью проводить тренировки в реальном ограниченном времени (codeforces.ru), с большим и разнообразным архивом олимпиад различного уровня (все упомянутые ресурсы).

5. Для проведения олимпиад школьного и муниципального уровня – olymp.isu.ru на основе АТС ejudge – из-за возможности прорешать или дорешать актуальные задачи недавних школьных и муниципальных уровней ВСоШ по информатике и ИКТ.

6. Для проведения олимпиад регионального уровня – acmp.ru, codeforces.ru, olymp.isu.ru на основе АТС ejudge, которые позволяют оперативно обновлять архивы задач регионально этапа, сопровождающие материалы и обсуждения

(acmp.ru, codeforces.ru), содержат инструменты тренировки навыков работы в актуальной среде тестирования (ejudge).

Заключение

При подведении общего итога можно заметить, что использование автоматизированных тестирующих систем с большим набором тем и задач во всех школах может поднять интерес к олимпиадному программированию и выявить талантливых учеников, которые не имеют пока возможности проявить себя в данной области.

Помимо олимпиадного направления подготовки данные ресурсы можно рекомендовать для обучения основам программирования всех учеников с помощью наборов типовых задач на условные операторы, циклы, массивы и подпрограммы. Очевидно, что данная методика имеет большие перспективы для базовой подготовки специалистов, деятельность которых будет существенно определять направления развития информационной сферы в будущем.

Заявленный вклад авторов

Зубков О. В.: *постановка проблемы, цели и задач исследования, практическая реализация разработанных методик, обоснование выбора определённых автоматизированных тестирующих систем на каждом этапе подготовки школьников.*

Семичева Н. Л.: *теоретическое обоснование актуальности исследования, систематизация полученной информации, проведение и обработка результатов педагогического эксперимента, оформление и редактирование статьи.*

Все авторы прочитали и одобрили окончательный вариант рукописи.

Список литературы

1. Абибуллаева А. О. Характеристика процесса развития алгоритмического мышления младших школьников на уроках математики // Проблемы современного педагогического образования. 2017. № 54-2. С. 10–15.

2. Баракина Т. В. Формирование у младших школьников элементов алгоритмической культуры на уроках информатики в начальной школе // Информатика в школе. 2014. № 7 (100). С. 58–61

3. Газейкина А. И. Стили мышления и обучение программированию студентов педагогического вуза [Электронный ресурс] // Информационные технологии в образовании : Конгресс конференций. ИТО-2006. Секция 1. Подсекция 1. URL: http://ito.edu.ru/2006/Moscow/I/1/I-1-6371.html (дата обращения: 23.06.2019).

4. Гаспарян А. В., Тимошина Н. В. Особенности автоматизации проверки задач по программированию [Электронный ресурс] // ИТпортал. 2018. № 2 (18). URL: http://itportal.ru/science/tech/osobennosti-avtomatizatsii-proverki/ (дата обращения: 02.07.2019).

5. Горчаков Л. В., Стась А. Н., Карташов Д. В. Обучение программированию с использованием системы Ejudge // Вестник Томского государственного педагогического университета. 2017. Вып. 9 (186). С. 109–112.

6. Долгушин Н. А., Оленькова М. Н. Использование системы Contester для проведе-

ния олимпиад по программированию [Электронный ресурс] // Международный студенческий научный вестник. 2016. № 3–2. URL: http://eduherald.ru/ru/article/view?id=14960 (дата обращения: 02.07.2019).

7. Еремеева Н. Н. Формирование алгоритмического мышления у школьников в ходе групповой работы // Пермский педагогический журнал. 2013. № 4. С. 86–89.

8. Канатьева Е. С., Мартынюк Ю. М. Понятие алгоритмического мышления [Электронный ресурс] // Научное сообщество студентов : Междисциплинарные исследования: электрон. сб. ст. по мат-лам XXII студенч. науч.-практ. конф. Новосибирск : АНС «СибАК». 2017. № 11(22). С. 161–165 (дата обращения: 02.07.2019).

9. Кодификатор элементов содержания и требований к уровню подготовки выпускников образовательных организаций для проведения единого государственного экзамена по информатике и ИКТ [Электронный ресурс] // утв. директором ФГБНУ «ФИПИ» 14 ноября 2018 г. М. : ФИПИ, 2018. 7 с. URL: http://fipi.ru/ege-i-gve-11/demoversii-specifikacii-kodifikatory (дата обращения: 02.12.2018).

10. Куликов С. Б. Опыт преподавания основ алгоритмики в средней школе // Педагогическое мастерство: мат-лы II Междунар. науч. конф. (г. Москва, 22–23 декабря 2012 г.). Магнитогорск : Буки-Веди, 2012. С. 115–117.

11. Лебедева С. Ю. Результаты государственной итоговой аттестации в форме единого государственного экзамена по информатике и ИКТ в Иркутской области в 2018 году. Методические рекомендации. Иркутск : ГАУ ДПО ИРО, 2018. 58 с.

12. Лебедева Т. Н. Пути формирования алгоритмического мышления школьников // Информатика и образование. 2008. № 6. С. 103–106.

13. Лучко Л. Г. Формирование алгоритмической культуры учащихся как системообразующая функция базового курса информатики [Электронный ресурс] // Информационные технологии в образовании : Конгресс конференций. ИТО-98. Секция 1 : устное выступление и публикация. URL: http://ito.edu.ru/1998/1/Luchko.html (дата обращения: 02.07.2019).

14. Николаева А. Д., Маркова О. И. Метапредметные компетенции как педагогическая категория [Электронный ресурс] // Современные проблемы науки и образования. 2015. № 4. С. 9. URL: http://www.science-education.ru/ru/article/view?id=20437 (дата обращения: 02.07.2019)

15. Чебурина О. В. Формирование алгоритмического мышления в обучении программированию игр [Электронный ресурс] // Наука и перспективы. 2017. № 2. URL: nip.esrae.ru/14-116 (дата обращения: 02.07.2019).

16. Bourouaieh D., Bensebaa T., Seridi H. Smart edutainment game for algorithmic thinking // Procedia – Social and Behavioral Sciences. 2012. Vol. 31. Pp. 454–458.

17. Hayal Y. M., Suheda Y. The Investigation of Algorithmic Thinking Skills of Fifth and Sixth Graders at a Theoretical Dimension // MATDER Journal of Mathematics Education. 2018. Vol. 3 No. 1. Pp. 41–48.

18. Knuth D. E. Algorithmic Thinking and Mathematical Thinking // The American Mathematical Monthly. 1985. Vol. 92. Issue. 3. Mar. Pp. 170–181. URL: https://doi.org/10.1080/00029890.1985.11971572.

19. Mezak J., Papak P. Pejic Learning scenarios and encouraging algorithmic thinking [Electronic resource] // Conference Paper 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) URL: https://www.researchgate.net/publication/326704371_Learning_scenarios_and_encouraging_algorithmic_thinking. DOI: 10.23919/MIPRO.2018.8400141 (mode of access: 14.09.2018).

The Role of Automated Testing Systems in Building Algorithmic Thinking Skills

Oleg V. Zubkov, Nataliya L. Semicheva
Irkutsk State University, Irkutsk

Abstract. ***Introduction.*** *The paper discusses the issues of building skills of algorithmic thinking in students in computer science lessons and elective programming classes. Modern teaching methods and automated testing systems designed to both develop basic skills and bring students to the highest level of programming skills are analyzed.*
Materials and methods. *The study is concerned with the problem of natural division of students during these classes into subgroups by level of training and specific features of thinking. The problem-solving approach using the automated testing systems is analyzed. A general review and classification of the most common resources with automatic program checking have been carried out.*
The results of the study. *This section shows the results obtained using the described methods and provides general recommendations for their distribution.*
Conclusion*. The proposed methods can be used in computer science lessons both for the general development of the algorithmic thinking of schoolchildren and for basic training of specialists whose activities will significantly determine the future development of the information sphere.*

Keywords: *algorithmic thinking, programming, small group work, automated testing systems.*

Зубков
Олег Владимирович

кандидат физико-математических наук, доцент, доцент кафедры алгебраических и информационных систем

https://orcid.org/0000-0002-1252-6486

Институт математики, экономики и информатики, Иркутский государственный университет

664003, Россия, г. Иркутск, б. Гагарина, 20

тел.: +7(3952)521277
e-mail: oleg.zubkov@mail.ru

Zubkov
Oleg Vladimirovich

Candidate of Sciences (Physical-Mathematical), Associate Professor at the Department of Algebraic and Information Systems

https://orcid.org/0000-0002-1252-6486

Institute of Mathematics, Economics and Computer Science, Irkutsk State University

20 Gagarin St, Irkutsk, Russia, 664003

tel.: +7(3952)521277
e-mail: oleg.zubkov@mail.ru

Семичева Наталья Леонидовна

кандидат физико-математических наук, доцент кафедры алгебраических и информационных систем

https://orcid.org/ 0000-0001-9306-9055

Институт математики, экономики и информатики, Иркутский государственный университет

664003, Россия, г. Иркутск, б. Гагарина, 20

тел.: +7(3952)521277
e-mail: natasliya@gmail.com

Semicheva Nataliya Leonidovna

Candidate of Sciences (Physical-Mathematical Sciences), Associate Professor at the Department of Algebraic and Information Systems

https://orcid.org/ 0000-0001-9306-9055

Institute of Mathematics, Economics and Computer Science, Irkutsk State University

20 Gagarin St, Irkutsk, Russia, 664003

tel.: +7(3952)521277
e-mail: natasliya@gmail.com