

**О. Ю. Заславская,**

**О. Д. Любутов**

## **Формы и методы подготовки школьников к олимпиадам по информатике с использованием электронных образовательных ресурсов**

В статье представлен опыт обоснования и проектирования электронного образовательного ресурса, способствующего эффективному обучению информатике и позволяющего совершенствовать системы подготовки школьников к участию в олимпиадах по информатике различного уровня. Описана методика обучения школьников решению олимпиадных задач по информатике, в которых для достижения результата целесообразно использование специфических структур данных (стека, очереди и дека).

Ключевые слова: методика обучения информатике; олимпиада по информатике; информационные технологии; алгоритмы.

**С**овременный мир переживает очередную научно-техническую революцию — революцию цифровизации общества. Необходимой составляющей перехода общества на повсеместное использование цифровых технологий является наличие у граждан информационной культуры, то есть совокупности знаний и умений, способствующих их эффективной информационной деятельности [4; 6; 7; 12]. Очевидно, что формирование информационной культуры граждан страны необходимо начинать еще со школы.

Существующая в России система поиска и выявления одаренных детей и талантливой молодежи нуждается в дополнительном развитии, считают в Экспертном совете при Правительстве РФ. В частности, необходимо совершенствовать систему предметных олимпиад и конкурсных процедур.

Важным шагом в систематизации и унификации поддержки талантливых детей стало утверждение в 2012 году Концепции общенациональной системы

выявления и развития молодых талантов, а также Стратегии развития и воспитания в Российской Федерации на период до 2025 года.

В свете этих тенденций учителя информатики ежегодно осуществляют подготовку школьников к участию в олимпиадах по информатике, причем, отметим, в условиях постоянного возрастания сложности олимпиадных заданий. На сегодняшний день эти задания на Всероссийской олимпиаде по информатике, начиная с муниципального этапа, далеко выходят за рамки базовой школьной программы [8; 10].

В настоящее время имеется достаточное количество электронных образовательных ресурсов (ЭОР), используемых для подготовки школьников к решению олимпиадных задач по информатике<sup>1</sup>.

Рассмотрим предназначение данных ЭОР с точки зрения изучения способов использования специфических структур данных (стек, очередь, дек) для решения олимпиадных задач по информатике.

Проведенный анализ существующих ЭОР для школьников, посвященных решению олимпиадных задач по информатике, показал: из семи выбранных ресурсов только один имеет встроенный справочник по структурам данных. Однако этот справочник реализован как электронная копия бумажного носителя, то есть он не может считаться полноценным ЭОР, так как не обладает ни мультимедийными возможностями, ни интерактивностью, что является неотъемлемыми качествами современных ЭОР.

Следовательно, существует потребность в создании специализированного ЭОР, предназначенного для изучения специфических структур данных (стека, очереди и дека), адресованного учащимся средней и старшей школы и обладающего мультимедийными возможностями, интерактивностью и интерфейсом, соответствующим уровню возрастных психофизиологических функций учащихся [1–3; 11].

С позиции дидактики учебный материал дисциплины следует отбирать в соответствии с определенными предметными и психолого-педагогическими требованиями, такими как целостность предоставления знаний, единство эмпирического и теоретического элементов обучения, полнота содержания курса, преемственность содержания.

Следует отметить, что в школьной программе содержание курса информатики в основном ориентировано на изучение программирования, систем

---

<sup>1</sup> Список наиболее популярных ЭОР: Информационный портал Всероссийской олимпиады школьников — [www.rosolymp.ru](http://www.rosolymp.ru); Олимпиады по информатике (Санкт-Петербург, Россия): официальный сайт Всероссийской командной олимпиады школьников по программированию — <http://neerc.ifmo.ru/school/>; Новости Олимпиады по программированию (соревнования по информатике и программированию, проводимые Новосибирским государственным университетом) — [olimpic.nsu.ru](http://olimpic.nsu.ru); Система проведения турниров по программированию Ejudge — <http://ejudge.ru/wiki/index.php/>; Timus Online Judge: архив задач по программированию с автоматической проверяющей системой — [acm.timus.ru](http://acm.timus.ru); Школа программиста — <http://www.acmp.ru/asp/do/index.asp>; Информатикс (дистанционная подготовка школьников по информатике) — [www.informatics.msk](http://www.informatics.msk).

счисления, кодирования информации и алгебры логики [5; 9]. Эти разделы являются фундаментальными в курсе информатики. В то же время в работах председателя Центральной предметно-методической комиссии по информатике В. М. Кирюхина [8: с. 38] указываются требования к знаниям и умениям участников олимпиад, которые либо вообще не рассматриваются, либо только упоминаются в программе курса информатики профильного уровня. В значительной степени это касается вопросов использования при решении олимпиадных задач по информатике специфических структур данных, таких как:

- стек, очередь, дек;
- дерево отрезков;
- бинарная куча;
- списки, словари и многое другое.

Для каждого из трех исполнителей (стек, очередь, дек) разработана линейка задач с постепенно возрастающей сложностью реализации. Способы решения отдельных задач вариативны, то есть допускают возможность использования различных структур данных. При этом существует вероятность того, что одна и та же задача будет проще реализовываться с использованием одной структуры данных (например, очереди) и сложнее — с использованием другой структуры (например, стека).

Соответствие предлагаемых задач классам сложности и типам структур данных представлено в таблице 1.

Таблица 1

#### Фрагмент классификации задач по классам сложности и типам структур данных

Сложности Задачи	Сложность 1			Сложность 2			Сложность 3		
	Стек	Очередь	Дек	Стек	Очередь	Дек	Стек	Очередь	Дек
Задача № 1	*	*			*				
Задача № 2	*				*				
Задача № 3		*							
Задача № 4						*		*	
Задача № 5						*		*	
Задача № 6				*	*				
Задача № 7								*	
Задача № 8	*	*	*						
Задача № 9	*								
Задача № 10		*							

В таблице 1 преобладают задачи с применением очереди. Это объясняется тем, что очередь является более универсальной структурой данных. Практически все задачи, решаемые с помощью стека, могут быть решены с помощью очереди, хотя и с меньшей степенью эффективности. И наоборот, многие задачи, решаемые с помощью очереди, не могут быть реализованы с помощью

стека. Дек является еще более универсальной структурой данных. Любая задача, решаемая с помощью стека или очереди, также может быть реализована с помощью дека. Поэтому в таблице 1 среди задач, решаемых с помощью дека, представлены задачи, которые могут быть решены с помощью дека и не могут быть решены с помощью стека или очереди, задачи, которые могут быть решены с помощью стека или очереди, но менее эффективно, чем с применением дека.

Ниже приведены некоторые примеры подобных задач и их решения (см. табл. 2, рис. 1 и рис. 2.).

Таблица 2

**Примеры задач, которые могут быть решены с помощью дека и не могут быть решены с помощью стека или очереди**

№	Условие задачи	Методические рекомендации по решению
1	<p>На стройплощадке имеются два склада. На первом складе расположен штабель (стопка) блоков одного цвета, различной яркости. Второй склад пуст. Конвейер работает в режиме стека или очереди. Требуется переместить штабель с первого склада на второй так, чтобы блоки располагались в порядке убывания яркости. Первый склад работает в режиме приема и выдачи, второй — в режиме приема товара</p>	<p>В данной задаче можно использовать различные алгоритмы сортировки. Например, сортировку выбором (метод локальных экстремумов). При помещении блоков с первого склада в стек, определяется максимальная яркость размещенных блоков. После чего блоки с максимальной яркостью помещаются на второй склад, а остальные блоки размещаются на первом складе (так как он работает и в режиме приема товара). После чего алгоритм повторяется для оставшихся на первом складе блоков. Аналогичный алгоритм применяется и при работе с очередью. При его реализации достаточно использовать два флага</p>
2	<p>На стройплощадке имеются два склада. На первом складе имеется штабель (стопка) блоков трех цветов (красного, желтого и зеленого), расположенных в случайном порядке. Второй склад пуст. Конвейер работает в режиме очереди или дека. Требуется переместить штабель с первого склада на второй так, чтобы расположение блоков по цветам соответствовало цветам светофора (сверху вниз: красный; желтый; зеленый). Первый склад работает в режиме выдачи, второй — в режиме приема товара</p>	<p>Алгоритм решения задачи следующий:</p> <ul style="list-style-type: none"> <li>– блок извлекается со склада;</li> <li>– если блок зеленого цвета, то он перемещается на второй склад.</li> </ul> <p>При использовании дека:</p> <ul style="list-style-type: none"> <li>– если блок желтого цвета, он помещается в левую часть дека, красный блок помещается в правую часть дека;</li> <li>– блоки извлекаются из левой части дека и помещаются на второй склад.</li> </ul> <p>При использовании очереди:</p> <ul style="list-style-type: none"> <li>– при помещении в очередь блоков желтого и красного цвета необходимо подсчитывать количество блоков желтого цвета;</li> <li>– из очереди извлекаются все блоки желтого цвета, затем все остальные (красные)</li> </ul>

Пример решения задачи № 1 с использованием стека представлен на рисунке 1.

Рассмотрим подробнее решение задачи № 1. Обстоятельство, что второй склад работает в режиме приема и выдачи товара, позволяет использовать его как вспомогательный буфер в алгоритме сортировки. В данной программе применен алгоритм сортировки, основанный на методе локальных экстремумов или выбора.

	ПРОГРАММА
1	КРАН ВПРАВО
2	ЦИКЛ
3	ПРИСВОИТЬ ФЛАГУ №1 ЗНАЧЕНИЕ 0
4	ПОКА ТЕКУЩИЙ СКЛАД НЕ ПУСТ
5	ВЗЯТЬ ГРУЗ
6	ПРИСВОИТЬ ФЛАГУ №2 ЯРКОСТЬ ГРУЗА
7	ЕСЛИ ФЛАГ №1 < ФЛАГ №2
8	ФЛАГ №1 <=> ФЛАГ №2
9	КОНЕЦ ЕСЛИ
10	КРАН ВЛЕВО
11	ПОСТАВИТЬ ГРУЗ
12	КРАН ВПРАВО
13	КОНЕЦ ПОКА
14	ПОКА БУФЕР КОНВЕЙЕРА НЕ ПУСТ
15	КРАН ВЛЕВО
16	ВЗЯТЬ ГРУЗ
17	КРАН ВПРАВО
18	ПРИСВОИТЬ ФЛАГУ №2 ЯРКОСТЬ ГРУЗА
19	ЕСЛИ ФЛАГ №1 = ФЛАГ №2
20	КРАН ВПРАВО
21	ПОСТАВИТЬ ГРУЗ
22	КРАН ВЛЕВО
23	ИНАЧЕ
24	ПОСТАВИТЬ ГРУЗ
25	КОНЕЦ ЕСЛИ
26	КОНЕЦ ПОКА
27	КЦ ЕСЛИ ТЕКУЩИЙ СКЛАД ПУСТ
28	

**Рис. 1.** Алгоритм решения задачи № 1 с использованием стека

В первой части программы все блоки с первого склада перемещаются в стек, при этом во флаге № 3 запоминается значение максимальной яркости перемещенных блоков. Затем все блоки извлекаются из стека. Если значение яркости очередного блока равно значению яркости, содержащемуся во флаге № 3, то данный блок помещается на второй склад, иначе — на первый склад. Таким образом, после первого прохода на втором складе окажутся все блоки с максимальной яркостью, а на первом складе — остальные блоки. После чего алгоритм повторяется до тех пор, пока на первом складе не останется ни одного блока. Пример решения задачи № 1 с использованием очереди представлен на рисунке 2.

ПРОГРАММА	
1	КРАН ВПРАВО
2	ПОКА ТЕКУЩИЙ СКЛАД НЕ ПУСТ
3	ПРИСВОИТЬ ФЛАГУ №2 ЗНАЧЕНИЕ 0
4	ПОКА ТЕКУЩИЙ СКЛАД НЕ ПУСТ
5	ВЗЯТЬ ГРУЗ
6	ПРИСВОИТЬ ФЛАГУ №1 ЯРКОСТЬ ГРУЗА
7	ЕСЛИ ФЛАГ №1 > ФЛАГ №2
8	ФЛАГ №1 <=> ФЛАГ №2
9	КОНЕЦ ЕСЛИ
10	КРАН ВЛЕВО
11	ПОСТАВИТЬ ГРУЗ
12	КРАН ВПРАВО
13	КОНЕЦ ПОКА
14	КРАН ВПРАВО
15	ПОКА БУФЕР КОНВЕЙЕРА НЕ ПУСТ
16	КРАН ВПРАВО
17	ВЗЯТЬ ГРУЗ
18	ПРИСВОИТЬ ФЛАГУ №1 ЯРКОСТЬ ГРУЗА
19	КРАН ВЛЕВО
20	ЕСЛИ ФЛАГ №1 = ФЛАГ №2
21	ПОСТАВИТЬ ГРУЗ
22	ИНАЧЕ
23	КРАН ВЛЕВО
24	ПОСТАВИТЬ ГРУЗ
25	КРАН ВПРАВО
26	КОНЕЦ ЕСЛИ
27	КОНЕЦ ПОКА
28	КРАН ВЛЕВО
29	КОНЕЦ ПОКА
30	

Рис. 2. Алгоритм решения задачи № 1 с использованием очереди

При использовании очереди алгоритм сортировки не претерпевает значительных изменений. Разница заключается лишь в том, что блоки помещаются и извлекаются с разных площадок. Поэтому обе эти задачи имеют средний уровень сложности. В случае, когда первый склад работает только в режиме выдачи товара (и не может быть использован в качестве вспомогательного буфера), сложность задачи возрастает. Она уже не может быть решена с помощью стека, но может быть решена с помощью дека или очереди. Эта задача повышенной сложности. Для ее реализации (особенно с помощью очереди) приходится задействовать все четыре флага. Но даже использование четырех флагов не позволяет провести алгоритм наиболее эффективно.

Отсутствие пятого флага заставляет использовать в каждом цикле лишнее циклическое перемещение блоков в очереди, что значительно увеличивает количество выполняемых команд.

По текстам обеих программ видно, что программа с использованием дека почти в полтора раза короче и использует только один флаг, в то время как программа с использованием очереди задействует два флага. Это происходит из-за того, что размещение блоков двух цветов в деке и в очереди носит различный характер. При помещении в дек блоки одного цвета размещаются с одной стороны, а другого цвета — с другой. В результате блоки в деке будут



отсортированы по цветам. Остается только извлечь все блоки с нужной стороны дека на склад.

При размещении в очереди блоки оказываются перемешаны по цветам. Чтобы извлечь из очереди блоки нужного цвета, необходимо поочередно извлекать из очереди блок, анализировать его цвет и в случае, если цвет нужный (в данной задаче это желтый цвет), поместить блок на второй склад, в противном случае необходимо поместить блок обратно в очередь. При этом следует постоянно контролировать количество блоков желтого цвета, которые еще не извлечены из очереди. Иначе алгоритм работы будет бесконечным. Именно для контроля количества желтых блоков в программе задействован флаг № 2. После того как все желтые блоки извлечены из очереди (флаг № 2 обнулился), из очереди извлекаются все остальные блоки (зеленого цвета) и помещаются на склад.

Подводя итог сказанному, можно заметить, что отбор содержательного наполнения ЭОР производился с учетом принципа систематичности и последовательности обучения, что важно для наиболее успешного усвоения материала обучающимися.

### Литература

1. Андреева Е. В. Программирование — это так просто, программирование — это так сложно. Современный учебник программирования. М.: МЦНМО, 2015. 184 с.
2. Ахо А. В., Хопкрофт Д. Э., Ульман Д. Д. Структуры данных и алгоритмы. М.: Вильямс, 2016. 384 с.
3. Вирт Н. Алгоритмы и структуры данных. Новая версия для Оберона. М.: ДМК Пресс, 2010. 272 с.
4. Гриншкун В. В., Заславская О. Ю., Корнилов В. С. Информационные технологии в образовании: учебно-методическое пособие. Воронеж: Научная кн., 2014. 70 с.
5. Заславская О. Ю. Подходы к формированию новых профессиональных качеств учителя информатики // Вестник Российского университета дружбы народов. Серия «Информатизация образования». 2008. № 3. С. 107–110.
6. Назарова Т. С. и др. Инструментальная дидактика: перспективные средства, среды, технологии обучения. М.—СПб., 2012. 278 с.
7. Заславская О. Ю., Филатова Н. И. Элементы теории управления в обеспечении углубленного обучения информатике // Вестник Российского университета дружбы народов. Серия «Информатизация образования». 2011. № 1. С. 49–55.
8. Кирюхин В. М. Информатика: всероссийские олимпиады. М.: Просвещение, 2008. 220 с.
9. Кузнецов А. А. и др. Информатика и ИКТ (Информационно-коммуникационные технологии). 8 класс: учебник для общеобразовательных учреждений. М.: Дрофа, 2010. 255 с.
10. Меньшиков Ф. Олимпиадные задачи по программированию. СПб.: Питер, 2006. 315 с.
11. Шень А. Х. Программирование. Теоремы и задачи. М.: МЦНМО, 2017. 320 с.

12. Kravets O. Ja., Zaslavskaya O. Ju. Adaptive management of individualizing computer science studies: patterns, algorithms, educational process. Yelm, WA, USA, 2014. 239 p.

### Literatura

1. Andreeva E. V. Programmirovaniye — e`to tak prosto, programmirovaniye — e`to tak slozhno. Sovremenny`j uchebnik programmirovaniya. M.: MCNMO, 2015. 184 s.

2. Axo A. V., Хоркрофт D. E`, Ul`man D. D. Struktury` danny`x i algoritmy`. M.: Vil`yams, 2016. 384 s.

3. Virt N. Algoritmy` i struktury` danny`x. Novaya versiya dlya Oberona. M.: DMK Press, 2010. 272 s.

4. Grinshkun V. V., Zaslavskaya O. Yu., Kornilov V. S. Informacionny`e tekhnologii v obrazovanii: uchebno-metodicheskoye posobie. Voronezh: Nauchnaya kn., 2014. 70 s.

5. Zaslavskaya O. Yu. Podxody` k formirovaniyu novy`x professional`ny`x kachestv uchitelya informatiki // Vestnik Rossijskogo universiteta družby` narodov. Seriya «Informatizaciya obrazovaniya». 2008. № 3. S. 107–110.

6. Nazarova T. S. i dr. Instrumental`naya didaktika: perspektivny`e sredstva, sredi`, tekhnologii obucheniya. M.–SPb., 2012. 278 s.

7. Zaslavskaya O. Yu., Filatova N. I. E`lementy` teorii upravleniya v obespechenii uglublennogo obucheniya informatike // Vestnik Rossijskogo universiteta družby narodov. Seriya «Informatizaciya obrazovaniya». 2011. № 1. S. 49–55.

8. Kiryuxin V. M. Informatika: vserossijskie olimpiady`. M.: Prosveshhenie, 2008. 220 s.

9. Kuznecov A. A. i dr. Informatika i IKT (Informacionno-kommunikacionny`e tekhnologii). 8 klass: uchebnik dlya obshheobrazovatel`ny`x uchrezhdenij. M.: Drofa, 2010. 255 s.

10. Men`shikov F. Olimpiadny`e zadachi po programmirovaniyu. SPb.: Piter, 2006. 315 s.

11. Shen` A. X. Programmirovaniye. Teoremy` i zadachi. M.: MCNMO, 2017. 320 s.

12. Kravets O. Ja., Zaslavskaya O. Ju. Adaptive management of individualizing computer science studies: patterns, algorithms, educational process. Yelm, WA, USA, 2014. 239 p.

**O. Yu. Zaslavskaya,**

**O. D. Lyubumov**

### **Forms and Methods of Preparing Students for Computer Science Olympiads using Electronic Educational Resources**

The article presents the experience of substantiating and designing a specially developed electronic educational resource that acts as an effective means of teaching computer science and allows improving the system of preparing students to participate in computer science Olympiads of various levels. The article describes a method of teaching students to solve Olympiad problems in computer science, in which it is advisable to use specific data structures (stack, queue, and deck).

Keywords: methods of teaching computer science; Olympiad in computer science; information technologies; algorithms.