

images/2019/01/obr-standarty-ob-sred-obrazovaniya.pdf . – Дата доступа: 01.04.2022.

5. Рындина, Ю. В. Исследовательская компетентность в структуре ключевых компетентностей будущего педагога [Электронный ресурс] / Ю. В. Рындина. – Режим доступа: <http://jurnal.org/articles/2011/ped4.html>. – Дата доступа: 28.03.2022.

6. Седеренко, П. В. Формирование готовности будущих педагогов к обучению учащихся исследовательским умениям и навыкам : автореф. дис. ... д-ра пед. наук : 13.00.08 / П. В. Седеренко ; МГПУ. – М., 2008. – 39 с.

7. Шефер, Е. А. Развитие профессиональной компетентности педагогов в области проектной и исследовательской деятельности с обучающимися / Е. А. Шефер // Школьная педагогика. – 2021. – № 2 (21). – С. 8–10.

ОРГАНИЗАЦИЯ ТРЕНЕРСКОЙ РАБОТЫ ПО ПОДГОТОВКЕ УЧАСТНИКА ОЛИМПИАДЫ ПО ИНФОРМАТИКЕ

*Буславский А. А., Белорусский
государственный университет, Республика
Беларусь*

На сегодняшний день никто не подвергает сомнению тот факт, что информационные технологии (ИТ) оказывают сильное влияние на жизнь современного человека. Возникают новые технологии, появляются новые сферы применения уже существующих технологий, появляются новые профессии, связанные с ИТ. Ситуация вызывает все возрастающий спрос на ИТ-специалистов, и поиск начинается уже не только среди студентов, но и среди школьников. Так как у учащихся могут быть разные таланты, способности и возможности к обучению, важно на раннем этапе выявить подходящих кандидатов на обучение и дать им возможность попробовать пройти учебный курс, получить специфические знания, умения и навыки. Конечно, информатика (англ. Computer Science, сокращенно CS) сейчас имеет много областей знаний, но наиболее наукоемкие из областей связаны с программированием и теорией алгоритмов [1].

Так сложилось, что информатикой называется также один из предметов общего среднего образования, наиболее близкий к CS. В качестве фундамента используются знания математики и логики. В разное время составители программы предмета «Информатика» делали различные акценты на содержание, от чистого программирования и

представления данных в компьютере, до формирования опытного пользователя информационных технологий. Сейчас существует баланс, при котором примерно треть курса посвящена обучению программированию, и две трети – умениям пользоваться прикладными программами и технологиями. Тем не менее международная олимпиада по информатике (англ. International Olympiad in Informatics, IOI) с самого начала включает только знания программирования и теории алгоритмов. Если сравнить области знаний и умений, требуемые от участника олимпиады [2] с требованиями к знаниям специалиста ИТ [1], то можно заметить большое количество совпадающих направлений, следовательно, подготовленного участника олимпиады по информатике можно считать ИТ-специалистом (пусть и начального уровня) в сферах программирования и теории алгоритмов. Участие в олимпиаде позволяет сравнить подготовку участников, продемонстрировать способность участников самостоятельно, быстро и эффективно решать незнакомые алгоритмические задачи. Международный характер олимпиады позволяет сравнить уровень подготовки учащихся (а в других соревнованиях и студентов) разных стран. Таким образом, подготовка победителей международных соревнований помогает поднять престиж национального образования и репутацию ИТ-специалистов своей страны.

В рамках данной статьи будет уделено внимание наиболее массовому этапу подготовки – начальному. На первых ступенях обучения многие темы рассматриваются вскользь, в ознакомительном режиме. На более высоких ступенях изучаются соответственно более эффективные (и более сложные для понимания и реализации) алгоритмы и структуры данных, задействуются новые знания, полученные при изучении математики. Многие темы могут изучаться повторно, с углублением. Рассмотрим подготовку участника олимпиады с точки зрения начинающего тренера (обычно это школьный учитель информатики). Целью тренера является подготовка победителя олимпиады высокого уровня, лучше всего международного.

Так как путь от новичка до победителя международной олимпиады долог и непрост, то проходят его далеко не все. Те, кто сошел с дистанции, получают знания и умения, позволяющие претендовать на роль ИТ-специалиста, и чем позже это случится, тем больший уровень знаний и умений будет получен. Если участник сумел в 11 классе получить диплом на заключительном этапе национальной олимпиады по информатике, то он получает возможность поступления без экзаменов в ведущие учреждения высшего образования страны, связанные с ИТ. Если нет, то

полученные знания и умения пригодятся в процессе обучения. Причины для ухода могут быть самые разные – переход в другое учреждение образование, серьезное увлечение другим предметом или деятельностью (спорт, танцы, музыка, рисование и т. п.), не оставляющее достаточно времени на соревновательную информатику. Если у тренера мало учеников, то можно остаться без учеников до достижения ими серьезных результатов. Чтобы избежать подобной ситуации, тренеру стоит позаботиться о достаточно большом количестве учеников на начальном этапе. С другой стороны, взять всех желающих обычно возможности нет. Значит, можно организовать первичный отбор, желательнее совмещая его с формированием нужных знаний и умений.

Как уже упоминалось, фундаментом для информатики является математика. Поэтому в раннем возрасте (младшая школа, иногда даже дошкольные учреждения) можно развивать логическое, математическое и алгоритмическое мышление. Одним из полезных инструментов является умение решать умные головоломки и играть в математические и логические игры. На данном этапе при разборе решения подобных задач (или определении выигрышной стратегии) важно научить учащегося читать условие задачи, выделять важные моменты, строить математическую модель, рассуждать, разделять задачу на подзадачи, доказывать правильность решения ([3], [4], [5], [6]). Большую помощь оказывает участие в онлайн-конкурсах, доступных в младшей школе ([14], [15]).

Следующим этапом в подготовке участника является обучение написанию программ на одном из языков программирования (кодирование алгоритма), поиску и исправлению ошибок (тестирование и отладка). В качестве основного языка участнику олимпиады по информатике имеет смысл выбрать C++ как единственный допущенный к IOI (а мы нацелены на участие в ней). В других олимпиадах допускаются также такие языки как Python, Pascal, Java, C# и пр. Но основными достоинствами C++ являются скорость, экономия памяти, наличие стандартных библиотек с большим количеством реализованных алгоритмов и структур данных, наличие бесплатных сред разработки (Codeblocks). Фактически многие разработчики олимпиадных заданий зачастую пишут авторские решения только на C++, не гарантируя наличие решений с такими же характеристиками по памяти и скорости на других языках программирования. Автор придерживается мнения, что сложность изучения C++ учащимися преувеличена, по крайней мере в той степени, которая позволяет решать олимпиадные задачи.

Так как олимпиада по информатике не предусматривает деления на классы, то имеет смысл начинать изучать программирование как можно раньше. Из опыта автора начинать обучение можно с 6–7 класса, а в отдельных случаях и раньше. В рамках занятий ЮНИ-Центра БГУ обучение программированию ведется в течение года по 2 академических часа в неделю (при этом подразумевается, что еще не менее 2 часов учащийся занимается дома самостоятельно). При этом за год учащиеся успевают изучить не только C++, но и Python (конечно, только в рамках необходимого для спортивного программирования, то есть, в частности, не изучается объектно-ориентированное программирование (ООП)). Второй язык может быть использован на олимпиадах, где он допущен, для решения подходящих задач (так, например, на Python проще обрабатывать строки и работать с длинными числами). Если начинающий тренер не обладает необходимыми знаниями и умениями программирования, то можно изучить язык самостоятельно по книгам [7], онлайн-курсам [8], интерактивным учебникам [9] или пройдя курс в одном из учебных центров. В рамках занятий ЮНИ-центра есть возможность учителю проходить учебный курс вместе со своими учащимися, закрепляя полученные знания и умения при выполнении домашних заданий. Наличие преподавателя позволяет оперативно решать возникающие проблемы. Для отработки навыков решения задач желательно использовать электронные задачки с автоматизированной системой проверки и учета выполнения задач ([10], [11]).

После того как участник научился писать и отлаживать программный код, можно приступить к изучению как готовых алгоритмов решения задач, так и методов разработки, анализа и оптимизации алгоритмов (будем называть изучаемый курс «Теория алгоритмов»). В рамках этого курса участник должен познакомиться с известными алгоритмическими задачами, изучить имеющийся опыт решения подобных задач. Важно не просто заучить готовые алгоритмы, а разобраться в составляющих алгоритма, научиться модифицировать его в зависимости от конкретной ситуации, обосновывать правильность алгоритма, выбирать подходящую под конкретные обстоятельства реализацию алгоритма. При этом объем изучаемого материала достаточно большой, и большинство учащихся может не успеть освоить его при работе по 2 часа в неделю (как и в случае с изучением программирования учащемуся рекомендуется дополнительно заниматься самостоятельно). В этом случае можно либо выделять больше часов на очную работу, либо предоставить обучающемуся на следующий год повторно пройти курс, закрепив

усвоенные темы и повторно изучив слабо изученные. Курс, с одной стороны, требует глубокого погружения в некоторые разделы математики, с другой стороны, практико-ориентированный, то есть требует нарезывания большого количества задач. Можно использовать готовые курсы, например, курс «Решение олимпиадных задач» [11]. Многие алгоритмы, изучаемые в данном курсе, могут оказаться сложными не только для учащегося, но и для тренера. В этом случае можно использовать как литературу (например, [12]), так и электронные библиотеки алгоритмов (например, [13]) или дистанционные онлайн-курсы. Как и в случае изучения языка программирования, можно посещать занятия учебного центра вместе с обучающимися. Опытных учащихся можно привлекать как к работе с начинающими, так и для разработки задач и проведения соревнований.

Нам видится, что становление тренера и последующее повышение его уровня будет проходить быстрее и безболезненнее при наличии учебных центров подготовки (далее – центров) на базе учреждений высшего образования, способных как предоставить преподавателей для проведения занятий, так и для консультирования или разработки олимпиадных задач. Кроме того, подобные центры могли бы взять на себя функции координаторов, собирая информацию о проводимых соревнованиях и олимпиадах по информатике (программированию) и донося ее до всех заинтересованных сторон. Также центры могли бы помогать в проведении сборов к крупным соревнованиям. Но даже при наличии таких центров с тренеров не снимается обязанность курирования своих учащихся, контролируя выполнение ими домашних заданий, диагностируя пробелы в знаниях и умениях и выработку путей для устранения недостатков подготовки. Нехватку квалифицированных кадров можно будет частично компенсировать, разделяя уровни кураторов. Так, непосредственно с учащимся будет работать его тренер, некоторые занятия будут проводить внешние преподаватели, также будут использоваться другие способы тренировки: изучение литературы, просмотр видеолекций, прохождение онлайн-курсов. С теми задачами, с которыми не будет справляться тренер, он может обращаться за помощью к региональному (районному, областному) куратору, который знает больше. В качестве таких кураторов могут, в частности, выступать бывшие успешные победители олимпиад. С теми задачами, с которыми не могут справиться региональные кураторы, помогут справиться специалисты центра. Сотрудники центра тоже могут нуждаться в обучении и могут быть направлены на повышение квалификации как внутри страны, так и за ее пределы. В настоящий

момент в качестве республиканского центра наиболее подготовленным выглядит ЮНИ-центр БГУ, из ближнего зарубежья – образовательный центр Сириус в России. В регионах могут назначаться наблюдатели, функциями которых будет не обучение, а учет и контроль решения задач и учебной активности тренеров и участников. По результатам наблюдений можно контролировать эффективность подготовки и принимать управляющие решения, в том числе осуществлять отбор (и по возможности, награждение) лучших участников.



Для формирования индивидуальной траектории подготовки каждого участника нужно будет отслеживать, что, как и в каком объеме он изучал, с каким тренером или источником работал и как закрепил изученное. Рабочий дневник может храниться в электронном виде, например, как электронная таблица. Чтобы не писать каждый раз длинное название источника, можно создать электронный каталог источников (книг, статей, сайтов, видеолекций и т. п.). Такой же каталог можно сделать и для электронных задачник. Чтобы можно было проконсультироваться по темам, недоступным текущему тренеру, нужно иметь список кураторов с указанием, какие темы куратор способен дать, в какое время, способ связи с куратором. Отдельно нужно учитывать результативность ученика по мере участия его в различных соревнованиях и тренировках. Такой дневник позволит не только отслеживать активность учащегося, но и отметить всех тренеров, которые принимали участие в его подготовке, начиная с младшей школы. Пример дневника можно посмотреть по ссылке: <https://docs.google.com/spreadsheets/d/1hmanpjs7oM3cV0G6nBW45Igm6gid9OPOxPD5dzo6NkM/edit?usp=sharing>.

Описанная система вполне может быть реализована и выглядит достаточно перспективно. Для ее организации и функционирования понадобится много временных и трудовых затрат всех заинтересованных сторон, но затраты должны окупиться уровнем

участников олимпиад и, как следствие, ростом профессиональной репутации страны и качества ее специалистов ИТ.

Список использованных источников

1. CS2023: ACM/IEEE-CS/AAAI Computer Science Curricula <https://csed.acm.org/>.
2. The International Olympiad in Informatics Syllabus <https://ioinformatics.org/files/ioi-syllabus-2023.pdf>.
3. Левитин, А. Алгоритмические головоломки [Электронный ресурс] / А. Левитин, М. Левитина ; пер. с англ. Ж. А. Меркуловой, Н. А. Меркулова. – 2-е изд. (эл.). – М. : Лаборатория знаний, 2019. – Электрон. текстовые дан. (1 файл pdf : 328 с.).
4. Перельман, Я. И. Веселые задачи / Я. И. Перельман. – М. : ООО «Издательство Астрель»: ООО «Издательство АСТ»: ООО «Транзиткнига», 2003.
5. Перельман, Я. И. Лабиринты / Я. И. Перельман. – 2-е изд. – Ленинград : Государственное издательство Москва, 1931.
6. Гарднер, М. Математические головоломки и развлечения / М. Гарднер. – 2-е изд., испр. и доп. – М. : Мир, 1999.
7. Лафоре, Р. Объектно-ориентированное программирование в C++. 4-е издание. / Р. Лафоре. – Питер, 2004.
8. Курс «Введение в программирование (C++)» [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/363/syllabus>.
9. Курс «Учите Питон» [Электронный ресурс]. – Режим доступа: <https://pythontutor.ru/>.
10. Система автоматического тестирования iRunner2 [Электронный ресурс]. – Режим доступа: <https://acm.bsu.by/>.
11. Сайт проекта «Школа программиста» [Электронный ресурс]. – Режим доступа: <https://acmp.ru/>.
12. Лааксонен, А. Олимпиадное программирование / А. Лааксонен ; пер. с англ. А. А. Слинкин / Антти Лааксонен. – М. : ДМК Пресс, 2018.
13. Библиотека алгоритмов MAximal [Электронный ресурс]. – Режим доступа: <https://e-maxx.ru/algo/>.
14. Международный конкурс по информатике «Бобер 2022» [Электронный ресурс]. – Режим доступа: <http://bebras.ru/>.
15. Международный математический конкурс «КЕНГУРУ» [Электронный ресурс]. – Режим доступа: <https://www.bakonkurs.by/kenguru/kenguru.php>.