

эффективных инструментов организации, проведения и проверки олимпиад и конкурсов по информатике» и многие другие.

В январе – марте 2022 г. совместно с Парком высоких технологий проведен Витебский областной открытый конкурс STEM-проектов для педагогических работников дошкольного и общего среднего образования, к участию в предварительном этапе которого были заявлены 322 работы педагогов по номинациям: «Дошкольник в мире STEM» – 151 работа; «STEM-педагог» – 171 работа (I–IV классы – 57 работ, V–VI классы – 35 работ, VII–VIII классы – 79 работ).

26 марта 2022 г. на базе Оршанской IT-академии «КОМПАС» состоялся финал конкурса. Результатом проведения конкурса стало создание педагогического сетевого сообщества «STEM-педагог: учитель будущего», которое открывает перед учителями возможности обмена актуальной информацией, использования открытых и свободных электронных ресурсов STEM-образования, самостоятельного создания сетевого учебного содержания, освоения информационных концепций, знаний и навыков STEM-образования, наблюдения за деятельностью участников сообщества, использования инновационных практик и распространения и популяризации педагогического опыта.

Опыт работы центра был представлен на XXVIII Международном форуме «ТИБО-2022», который проходил с 6 по 10 июня 2022 г. в г. Минске.

Таким образом, следуя основным принципам STEM-образования, центр обучает педагогов Витебской области, координирует их работу. Концептуальные основы деятельности центра определяются высказыванием Майкла Окино, профессора образовательной школы Уорнера университета Рочестера: «Чтобы знания запоминались, надо их проживать... Учитель, сам ни разу не проводивший исследовательскую работу, не может научить исследованиям и детей...». Сегодня региональный центр педагогического STEM-образования – это платформа и ресурсная площадка, где концентрируются новые разработки, модели, которые становятся доступными для изучения и применения в образовательном процессе, а также для активных коммуникаций, обмена опытом, рефлексии и новых поисков.

Таким образом, будущее за технологиями, а будущее технологий за учителями нового формата.

Буславский А. А. (г. Минск, Республика Беларусь)

ПОДГОТОВКА УЧИТЕЛЯ КАК ТРЕНЕРА УЧАСТНИКА ОЛИМПИАДЫ ПО ИНФОРМАТИКЕ

Существует много книг и статей о том, как готовить учащегося к участию в олимпиаде по информатике. В то же время наблюдается дефицит тренеров, ученики которых регулярно показывают высокий результат на значимых олимпиадах. В рамках данной статьи рассмотрим возможный путь формирования тренера с момента возникновения желания заняться такой деятельностью.

Работа по подготовке участника начинается в младшем возрасте и продолжается вплоть до окончания высшего учебного заведения. При этом у участника может быть как один тренер на весь возрастной диапазон, так и ряд тренеров, каждый из которых занимается своей частью работы. Рассмотрим подготовку участника по этапам:

1. Младшая школа.

Как ни странно, начинать нужно не с изучения языка программирования, хотя это, безусловно, обязательное и важное умение. Но не менее важным является необходимость как можно ранее выявить ученика, способного думать, рассуждать, находить закономерности, уметь работать самостоятельно, обладать соревновательными навыками. При этом возраст ученика может быть небольшим – вплоть до младшей школы, а в некоторых отношениях – и до детского сада.

Безусловно, возраст накладывает свои ограничения – есть специфика работы с детьми младшего возраста. Но тем не менее навыки, полученные в юном возрасте, увеличивают шансы участника на успех в соревнованиях. В настоящее время, насколько известно автору, нет понятия тренера для младшей школы, но для системной работы его роль важна и значима.

Одной из причин отсутствия тренеров для младшей школы видится в системе поощрения учителей, работающих с одаренными детьми. Тренер получает вознаграждение (премии, надбавки, освобождение от общественной работы или классного руководства) только в случае получения дипломов на олимпиадах высокого уровня, таких как национальная предметная олимпиада или олимпиада международного уровня (IOI, EJOI, EGOI, Жаутыковская, ISIJ и т. п.). При этом у ученика младшей школы практически нет шансов такой диплом заработать, ведь по информатике задачи обычно даются одни и те же для всех возрастных категорий и приходится соревноваться с более старшими участниками, имеющими больший стаж подготовки и лучшую математическую базу, что определяет невысокую вероятность получения диплома. Можно предположить, что именно отсутствие практической возможности подтвердить результативность подготовки участника дипломом отталкивает потенциальных тренеров.

Но на начальном этапе важно не получить диплом, а построить базу, на которой будет строиться будущий результат – умение работать самостоятельно, планировать свою деятельность, контролировать и оценивать уровень владения инструментами, работать над ошибками. Для тренера важно научиться индивидуальной работе с участником, учитывая его особенности, сильные и слабые стороны.

Ситуацию можно было бы изменить к лучшему, если бы система поощрений работала на всех тренеров, работавших с результативным участником. Также частичным решением проблемы является организация интеллектуальных соревнований для участников младшего возраста и учет результатов участников в системе поощрений тренеров. При этом не все соревнования для младших школьников одинаково влияют на развитие необходимых качеств. Например, для олимпиады по информатике в большей степени полезны конкурсы, развивающие логику, математическую базу и алгоритмическое мышление, такие как «Бобер», «Scratch», «Кенгуру». В меньшей степени влияют конкурсы, проверяющие общие знания по предмету или практические навыки использования информационных технологий, такие как «Информышка», создание презентаций, сайтов, видеороликов и т. п. Будущему участнику олимпиады по информатике очень полезно принимать участие в соревнованиях по математике для младших классов, помогающих сформировать логическое, алгоритмическое и математическое мышление. Полезно решение интеллектуальных головоломок (например, расстановка ферзей, Ханойские башни, магический квадрат и др.), увлечение логическими и математическими играми («Крестики-нолики», «Быки и коровы» и др.). Проведение соревнований по таким играм и разбор выигрышных стратегий могут привлечь способных учащихся к дальнейшему росту уже как участников олимпиад. В качестве примера литературы по теме можно привести книгу «Алгоритмические головоломки» [1].

Существуют факультативы для младшей школы по информатике, но на текущий момент на сайте НИО РБ размещен только факультатив по Scratch [2] (в основном для формирования алгоритмического мышления (полезный навык для олимпиадника) и введения в объектно-ориентированное программирование (в олимпиадах не используется)). Впрочем, есть интересные факультативы по математике [3], которые могут помочь сформировать навыки математического мышления и логики. Будущему тренеру полезно изучить литературу по теме, существующие программы

факультативов (не только отечественных), пройти повышение квалификации (такие курсы или семинары могли бы проводиться на базе института развития образования или академии последипломного образования), но на текущий момент автору известен только цикл семинаров для учителей на базе ЮНИ-центра-XXI. Для учеников младшей школы проводятся занятия в течение года на базе ЮНИ-центра. Кроме того, проводятся занятия в Республиканской летней научно-исследовательской школе на базе спортивно-оздоровительного комплекса «Бригантина» БГУ, Международной летней онлайн-школе “Mathematics is my life”. Можно принять участие в межрегиональной многопрофильной олимпиаде по направлениям «Математика» и «Информатика». Более подробную информацию можно получить на сайте uni.bsu.by.

2. Средняя школа – начальный уровень (программирование).

Начиная с V-VI класса можно заниматься изучением языка программирования. В отдельных случаях можно и раньше (индивидуально), по факту большинство учащихся начинает заниматься программированием с VII класса. Выбор языка программирования с одной стороны не так важен (важно умение переводить алгоритм в команды языка программирования), с другой стороны, опытные участники олимпиад по информатике используют только C++. Выбор этого языка обусловлен его высокой скоростью и эффективностью, наличием мощных инструментов для решения алгоритмических задач (в первую очередь, Standart Template Library (STL), библиотека `algorithm`), наличием общедоступной, малотребовательной по ресурсам компьютера, многоплатформенной среды разработки (CodeBlocks). Именно поэтому этот язык является единственным выбором для участника международной олимпиады по информатике (IOI). На других олимпиадах по информатике доступны и такие языки программирования, как Pascal, Python, Java и др., но по разным причинам они уступают языку C++. Поэтому появляется выбор: начинать обучение программированию с C++ или переходить на него с другого языка программирования (который выбирается с точки зрения простоты обучения или присутствия описания в учебнике по информатике). Да, существует такая точка зрения, что начинать учиться программированию нужно с простого языка, например Pascal или Python. Но из практики работы с учениками автор сделал вывод, что при обучении с нуля не так важно, каким командам учиться (Pascal или C++), для начинающего ученика они одинаково непонятны, но в Pascal нет готовых библиотек алгоритмов и структур данных, а Python не дает, например, таких важных знаний, как строгая типизация, различие между переменной и указателем, возможность работы со строкой как с массивом. Кроме того, такие языки, как Python, C#, Java, работают медленно из-за необходимости использования виртуальной машины, что не дает возможности решить любую задачу в предлагаемых автором ограничениях, хотя процесс разработки может быть более быстрым, чем на C++ (так, например, на Python гораздо удобнее работать со строками). Если ученик не планирует участвовать в олимпиадах, то можно изучать любой язык программирования (даже на уроках информатики, это не противоречит нормативным документам РБ, только необходимо предоставить ученику учебный и справочный материал для изучения языка). В истории предмета информатики в РБ это был и Basic, и алгоритмический язык (Кумир), и Pascal (Boland, Free, ABC.NET). В новых версиях учебника, которые войдут в учебный процесс с 2023 года, будет Python 3. Большинство учителей идет по легкому пути и в рамках предмета «Информатика» дают тот язык, который приведен в учебнике. Но есть и те, кто уже много лет дает другие языки (например, C++, что упрощает подготовку олимпиадников, давая при этом остальным учащимся знание профессионального языка программирования, являющегося базой для большинства современных языков программирования). Для того чтобы изучить язык программирования, не описанный в

учебнике, можно использовать литературу (для C++ в качестве одной из наиболее понятных и полных книг для новичка можно упомянуть «Объектно-ориентированное программирование в C++» [4]. Можно пройти бесплатный онлайн-курс, например, на stepik.org [5]. Если предпочтение отдано языку Python, есть хороший электронный учебник pythontutor.ru [6]. Кроме того, можно пройти курс на базе одного из учреждений повышения квалификации. Еще одним способом выучить язык, а заодно посмотреть на методику его преподавания – записаться в качестве участника курса ЮНИ-центра-XXI вместе со своими учениками. Одним из недостатков большинства книг или дистанционных курсов является небольшое количество задач, или задачи, требующие хорошего знания математики (многие онлайн-курсы ориентированы на взрослую аудиторию). Восполнить этот пробел можно онлайн-задачниками, на которых задачи по программированию проверяются автоматизированными системами. Одним из лучших ресурсов является aspr.ru [7]. На этом сайте можно пройти курс «Обучение программированию на C++», причем сдавать задачи (более 200 задач) можно на любом из имеющихся в системе языков (более 10, включая C++, Python, Pascal, Java и др.). Большинство задач курса можно решить в течение одного учебного года. Также можно организовывать свои учебные группы и отслеживать решения учеников. В интернете можно найти видеоразборы отдельных задач курса. Тренеру рекомендуется самому пройти курс, чтобы иметь возможность быстро и квалифицированно помочь своим ученикам. Необходимо разъяснять учащимся, что использование чужих решений вредно, так как не позволяет понять, где есть пробелы в знаниях. В рамках занятий ЮНИ-центра у преподавателей есть разные подходы к темпам и объемам изучения языка программирования. Автор уже несколько лет практикует вариант, в котором в течение учебного года изучается два языка программирования (сначала C++, затем Python), с использованием в качестве задачника aspr.ru. Возраст участников групп – от V до X класса. По результатам учебного года можно перейти в группу, изучающую теорию алгоритмов.

3. Средняя и старшая школа – продолжающие (теория алгоритмов).

Изучив один или несколько языков программирования, ученик получает инструмент для создания различных приложений и может двигаться дальше по разным направлениям. Можно писать программы для роботов (C++, Python) и участвовать в конкурсах по робототехнике. Можно переключиться на разработку игр как на профессиональных игровых движках (Unreal Engine – C++, Unity – C#), так и на мобильных платформах (iOS - C, Android – Java, Python). Можно специализироваться по разработке сайтов (HTML+CSS+JS+PHP, Python), баз данных (SQL, C#, Python), машинному обучению (Python), тестированию, автоматизации документооборота, управлению серверами, кибербезопасности, блокчейн и многому другому. В рамках этих направлений можно создавать факультативы и курсы по выбору. Это все также прекрасно подойдет для участия в конкурсах работ исследовательского характера. Но в рамках работы ЮНИ-центра было решено заострить внимание на теории алгоритмов как наиболее наукоемком разделе информатики, доступного школьнику. Также это направление напрямую используется для решения сложных алгоритмических задач, в том числе задач по информатике, поэтому тренер участника олимпиады с этим направлением должен быть знаком очень хорошо.

Итак, есть ученик, обладающий навыками логики, алгоритмизации, математическим мышлением, умеющий программировать на одном из языков программирования (предпочтительно C++). Он уже может решить многие олимпиадные задачи, но выведение решения с нуля занимает слишком много времени. Поэтому нужно, во-первых, познакомить его с методами разработки алгоритмов, во-вторых, с наиболее известными и используемыми алгоритмами и, учитывая, что есть

много способов решить задачу, научить правильно выбирать алгоритмы и структуры данных в каждом конкретном случае. Для простоты разобьем алгоритмы по темам, которые можно изучать в некотором порядке. Различаются способы разбиения алгоритмов на группы, порядок их изучения, но в целом придерживаются правила «от простого к сложному», то есть более сложная задача может основываться на предварительном решении простых задач. Деление автором задач на темы в рамках изучения курса теории алгоритмов: целочисленная арифметика, поиск и сортировка, комбинаторика, перебор, структуры данных, рекуррентные соотношения, теория графов, геометрия на плоскости. Содержание курса во многом совпадает с курсом «Решение олимпиадных задач» на aspr.ru, что позволяет использовать этот курс в качестве задачника с автоматизированной проверкой. Также содержание курса во многом совпадает с курсом «Теория алгоритмов» для 2-го курса факультета прикладной математики и информатики Белорусского государственного университета. Безусловно, есть отличия, так как студентам дается сложная математическая база для оценки сложности алгоритмов и доказательства его правильности, что сложно дать школьникам. Есть различия и в содержании. Например, школьникам не рассказывается о поисковых деревьях, а студентам – об алгоритмах на строках. Есть различия в сроках прохождения курса: студенты проходят его в течение семестра, а школьники – как минимум в течение года (не многие успевают все усвоить за год, в таком случае можно повторно пройти курс, повторив известные темы и изучив те, которые не были усвоены ранее). Закреплять изученные алгоритмы необходимо решением задач на соответствующие темы.

Помимо вышеупомянутого курса на сайте aspr.ru выложены и доступны к решению десятки олимпиад разной сложности. Для тех учеников, кому уровня задач этого сайта уже недостаточно, можно предложить сайт codeforces.com [8], на котором еженедельно проходят олимпиады по программированию. Кроме того, тренером Алексеем Боруновым на базе yandex.contest создана площадка [9], на которой размещены национальные олимпиады прошлых лет и открытые тренировки. Есть электронные сборники статей по алгоритмам, из которых наиболее известен e-maxx.ru [10]. Есть циклы видеолекций на youtube. Из литературы для начинающих можно рекомендовать книги В.М. Котова [11, 12], М. С. Долинского [13, 14], С. М. Окулова [15, 16], А. Шеня [17]. Для тех, чей основной язык программирования C++, подойдет сборник «Фундаментальные алгоритмы на C++» [18, 19]. В книге «Методика решения задач по информатике» [20] рассматривается решение задач международных олимпиад по информатике (IOI). Для олимпиадников–международников настольной стала книга «Алгоритмы: построение и анализ» [21]. Из появившихся недавно книг можно упомянуть «Олимпиадное программирование» [22] и «Спортивное программирование» [23].

Итак, мы прошли большой путь. В сотрудничестве с учителями младших классов и математики провели подготовительную работу в младшей школе. За год-два обучили учеников программированию на одном или нескольких языках программирования. За два-три года изучили с учениками основные алгоритмы и структуры данных и отточили умение их использовать на олимпиадных задачах различных олимпиад. Настала пора получить результат нашего труда, отправив учеников в качестве участников олимпиады по программированию. Олимпиады есть как в нашей стране, так и за ее пределами, как очные, так и дистанционные, как индивидуальные, так и командные, как по одному предмету, так и по их сочетаниям. Помимо потенциального получения наград ученик должен постоянно самосовершенствоваться, используя в качестве индикатора результаты участия в соревновании. Задачей тренера является построение наиболее эффективной программы

обучения и графика тренировок. Начинают становиться важными такие факторы, как умение читать условие задачи, скорость набора текста, грамотное распределение времени на задачи, психологическая устойчивость и уверенность в себе.

Список использованных источников

1. Левитин, А. Алгоритмические головоломки. – 2-е изд. / А. Левитин, М. Левитина ; пер. с англ. Ж. А. Меркуловой, Н. А. Меркулова. – М. : Лаборатория знаний, 2019. – 325 с. : ил.
2. Учебная программа факультативного занятия «Творческая деятельность в среде программирования Scratch» для учащихся II–IV классов учреждений образования, реализующих образовательные программы общего среднего образования [Электронный ресурс]. – Режим доступа : <https://adu.by/images/2021/08/up-tv-deyat-v-srede-program-Scratch-2-4kl.pdf>. – Дата доступа : 22.06.22.
3. Учебные программы факультативных занятий по учебным предметам 2021/2022, IV класс [Электронный ресурс]. – Режим доступа : <https://adu.by/ru/homepage/obrazovatelnyj-protsess-2021-2022-uchebnyj-god/obshchee-srednee-obrazovanie-2021-2022/303-uchebnye-predmety-i-iv-klassy-2020-2021/3801-iv-klass.html>. – Дата доступа : 22.06.22.
4. Лафоре, Р. Объектно-ориентированное программирование в C++. – 4-е изд. / Р. Лафоре. – СПб. : Питер, 2018. – 928 с. : ил.
5. Каталог учебных курсов [Электронный ресурс]. – Режим доступа : <https://stepik.org/learn>. – Дата доступа : 22.06.22.
6. Интерактивный учебник языка Питон [Электронный ресурс]. – Режим доступа : <https://pythontutor.ru/>. – Дата доступа : 22.06.22.
7. Школа программиста [Электронный ресурс]. – Режим доступа : <https://acmp.ru/>. – Дата доступа : 22.06.22.
8. Соревнования по программированию 2.0 [Электронный ресурс]. – Режим доступа : <https://codeforces.com/>. – Дата доступа : 22.06.22.
9. Клуб юных пожарных: тренировочные олимпиады [Электронный ресурс]. – Режим доступа : <https://official.contest.yandex.ru/CYF/news>. – Дата доступа : 22.06.22.
10. Открыта библиотека алгоритмов [Электронный ресурс]. – Режим доступа : <https://e-maxx.ru/algo/>. – Дата доступа : 22.06.22.
11. Котов, В. М. Информатика. Методы алгоритмизации : учеб. пособие для 8-9-х кл. общеобразоват. шк. с углубл. изучением информатики с рус. языком обучения / В. М. Котов, И. А. Волков, А. И. Лапо. – Минск : Нар. асвета, 2000. – 300 с. : ил.
12. Котов, В. М. Информатика. Методы алгоритмизации : учеб. пособие для 10-11-х кл. общеобразоват. шк. с углубл. изучением информатики / В. М. Котов, О. И. Мельников. – Минск : Нар. асвета, 2000. – 221 с. : ил.
13. Долинский, М. С. Алгоритмизация и программирование на Turbo Pascal: от простых до олимпиадных задач : учебное пособие / М. С. Долинский. – СПб. : Питер, 2005. – 237 с. : ил.
14. Долинский, М. С. Решение сложных и олимпиадных задач по программированию : учебное пособие / М. С. Долинский. – СПб. : Питер, 2006. – 366 с. : ил.
15. Окулов, С. М. Основы программирования / С. М. Окулов. – М. : БИНОМ. Лаборатория знаний, 2004. – 424 с. : ил.
16. Окулов, С. М. Программирование в алгоритмах / С. М. Окулов. – М. : БИНОМ. Лаборатория знаний, 2002. – 341 с. : ил.
17. Шень, А. Программирование: теоремы и задачи. – 6-е изд., дополненное / А. Шень. – М. : МЦНМО, 2017. – 320 с. : ил.

18. Седжвик, Роберт. Фундаментальные алгоритмы на С. Анализ. Структуры данных. Сортировка. Поиск : пер. с англ. / Роберт Седжвик. – СПб : ДиаСофтЮП, 2003. – 672 с.
19. Седжвик, Роберт. Фундаментальные алгоритмы на С. Алгоритмы на графах : пер. с англ. / Роберт Седжвик. – СПб : ДиаСофтЮП, 2003. – 480 с.
20. Кирюхин, В. М. Методика решения задач по информатике. Международные олимпиады / В. М. Кирюхин, С. М. Окулов. – М. : БИНОМ. Лаборатория знаний, 2007. – 600 с. : ил.
21. Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – Москва : МЦНМО, 2000. – 960 с., 263 ил.
22. Лааксонен, А. Олимпиадное программирование. – 2-е изд., обновленное и дополненное / пер. с англ. А. А. Слинкин. – М. : ДМК Пресс, 2020. – 328 с. : ил.
23. Халим, С. Спортивное программирование / С. Халим, Ф. Халим ; пер. с англ. Н. Б. Желновой, А. В. Снастина. – М. : ДМК Пресс, 2020. – 604 с. : ил.

СОДЕРЖАНИЕ И МЕТОДОЛОГИЯ

Гундина М. А., Князев М. А., Крушевский Е. А.
(г. Минск, Республика Беларусь)

РАЗВИТИЕ КРЕАТИВНОГО ПОДХОДА В ПРЕПОДАВАНИИ МАТЕМАТИКИ БУДУЩИМ ИНЖЕНЕРАМ

С использованием информационных технологий находят свое применение в преподавании в вузах. Лекция в виде презентации с пояснениями преподавателя давно стала нормой. Практические занятия (например, по математике) с использованием собственного устройства (компьютера или просто смартфона) для решения математических и инженерных задач уже не кажутся фантазией. Стал возможным прием экзамена в электронной форме. При новом типе оценивания студенты сдают экзамен на компьютере или даже смартфоне. Такие экзамены приходят на смену классическим формам организации как промежуточного, так и итогового контроля знаний обучающихся. Подобные подходы также широко применяются и в зарубежных университетах [1, 2].

Развитие этого направления в организации учебного процесса требует разработки нового подхода к проведению лекционных, практических и лабораторных занятий. ЭУМК приходят на смену классическим учебно-методическим пособиям, электронные формы заменяют стандартные бланки ответов на экзамене [3]. Использование интерактивных математических приложений *MathLab* и *Mathematica Cloud* делает доступным работу над исследованиями в любом месте в любое время.

Современного студента следует воспринимать в неразрывной связи с миром информационных и телекоммуникационных технологий. На примере технического университета применение информационных технологий в учебном процессе неразрывно связано с уровнем сформированности профессиональных компетенций у будущих инженеров, таких как проектирование и разработка программных инженерных решений; анализ программных инженерных задач; оценка инженерной деятельности; при помощи средств вычислительной техники и программного обеспечения разработка технологии решения задач по обработке информации; определение структуры информации, схемы ее ввода и обработки; осуществление сопровождение использования готовых программных продуктов; разработка инструкции по работе с компьютерными программами; оформление необходимой технической документации и др. [2].

Рассмотрим для начала трансформацию такого вида занятий, как лекция. Если раньше единственно возможным вариантом была доска, мел и преподаватель, который