

УДК 004

## ОБ ОТКРЫТОМ ЧЕМПИОНАТЕ СГУ ПО ПРОГРАММИРОВАНИЮ

*Н. О. Котелина, Н. К. Попова, М. Н. Юркина*

Статья посвящена юбилейному XX Открытому чемпионату Сыктывкарского государственного университета по программированию. Рассказывается об опыте проведения мероприятия, а также о лицах, внесших значительный вклад в олимпиадное движение.

*Ключевые слова:* спортивное программирование, ACM, ICPC.

### **1. О студенческом чемпионате мира по программированию**

Международная студенческая олимпиада по программированию (в английском принято сокращение ACM/ICPC или просто ICPC) известна также как командный студенческий чемпионат мира по программированию<sup>1</sup>. Среди сборных команд высших учебных заведений чемпионат проводится с 1977 года. Его организатор, Международная ассоциация ACM (Association for Computing Machinery<sup>2</sup>), входит в тройку наиболее авторитетных ассоциаций ИТ-профессионалов в мире. Отбор на финальные соревнования чемпионата проводится по многоступенчатой схеме: все университеты мира разбиты по территориальному признаку на регионы, в которых проводятся полуфиналы. Наиболее крупные регионы разбиты на зоны, и отбор в полуфинал проводится по результатам четвертьфиналов.

Сыктывкарский государственный университет имени Питирима Сорочкина (далее СГУ) входит в Северо-восточную зону с центром в Екатеринбурге. С 2015 года четвертьфинальные соревнования проходят в два тура. Первый (квалификационный) тур проходит на площадках университетов в режиме online. В этом году в квалификационном туре приняли

<sup>1</sup>Официальная страница олимпиады. URL: <https://icpc.baylor.edu/>.

<sup>2</sup><http://www.acm.org>.

участие 428 студенческих команд, представляющие 68 учебных заведений из 27 городов на 20 площадках. В СГУ в квалификационном туре участвовали 17 команд.

СГУ участвует в мировом олимпиадном движении с 1999 года, когда команда в составе студентов кафедры прикладной математики Зелянина Романа, Подорова Максима, Чупрова Владимира заняла 9 место в четвертьфинале (г. Екатеринбург) и была участницей полуфинала командного чемпионата мира по программированию в Санкт-Петербурге, где вошла в число квалифицированных команд, заняв 38 место из 88 команд. Этот год мы и считаем началом Открытого чемпионата СГУ по программированию, хотя он так еще не назывался и его проведение приказами не оформлялось. В 2001 году кафедрой прикладной математики был организован и проведен внутривузовский чемпионат по программированию в три тура: два личных тура и заключительный командный. В следующие годы формат неоднократно менялся. В 2014 году первый личный тур стал проводиться как «Первокурсник», с отдельным подведением итогов среди студентов первых курсов университета. Начиная с 2018 года кафедрой проводится также интернет-тур [1].

## 2. О чемпионате с точки зрения участника. Правила

Главной особенностью международной студенческой олимпиады по программированию является то, что она командная. Правила соревнований ICPC сводятся к следующему.

- Состязания командные: команда состоит из трех человек и на каждую команду выделяется один компьютер. Существуют правила, ограничивающие состав участников по возрасту и т. п.
- Всем командам предъявляется один и тот же набор задач, обычно от 8 до 12. Продолжительность соревнования 5 часов. Официальный язык — английский. При предварительном отборе на квалификационных турах допускается использование национального языка, например в нашей стране — русского. Решением задачи является программа.
- Программы тестируются на большом количестве различных входных тестов, неизвестных участникам. Тестирование выполняется до первого теста с неправильным ответом. Если программа выдала неправильный ответ или не уложилась в ограничения по времени или памяти, то пославшая ее команда получает об этом сообщение и может послать исправленную версию. Но каждая новая попытка штрафуетсь добавочным временем.

- Задача считается решенной, если программа выдала правильные ответы на всех тестах. Турнирное положение команд рассчитывается исходя из количества решенных задач и полученного штрафного времени (за неудачные попытки). В штрафное время включается также время, прошедшее от начала соревнования до момента признания решения задачи правильным. При равном количестве решенных задач лучшее место имеет команда с наименьшим штрафным временем.

Участники и болельщики могут видеть текущее состояние турнирной таблицы. В таблице команды упорядочены по количеству решенных задач и штрафному времени, есть информация о решенных каждой командой задачах, о количестве попыток. Отмечаются команды, первыми решившие какую-либо задачу. Существование турнирной таблицы добавляет соревнованиям зрелищности, а участникам и болельщикам эмоциональных переживаний. Существует традиция награждать каждую команду за каждую решенную задачу воздушным шариком определенного цвета. Это делает обстановку на соревновании по-настоящему праздничной.

Правила неоднократно критиковались за оторванность от реальной программистской деятельности. Например, все задачи оцениваются одинаково. Фактически это означает, что если в блоке задач была одна трудная и несколько простых, то команда, решившая две простые задачи, займет более высокое положение, чем команда, которая решила одну, но трудную задачу.

Решения оцениваются по работе программ, их представляющих. Это неправильно в принципе: еще Дейкстра писал о том, что тестирование не может доказать корректность работы программы, оно может лишь опровергнуть это утверждение. Таким образом, отказ от анализа решения приводит к тому, что некоторые участники стараются сделать программу, дающую ответы «как у жюри» на предложенные тесты, а не решить задачу. Тем не менее, нельзя отрицать, что тестирование — это объективная мера для определения лучших участников, а процедура определения победителей прозрачна и открыта.

За прошедшие десятилетия правила соревнований АСМ неоднократно изменялись и в настоящий момент направлены скорее на поддержание спортивного азарта и «зрелищности» соревнований.

### **3. О чемпионате с точки зрения организаторов**

Олимпиады с математико-программистским содержанием обычно называют олимпиадами по информатике (например, ВсОИ) или олим-

пиадами по программированию (ICPC). Под программированием в узком смысле слова обычно понимается запись алгоритма на языке программирования. Однако в олимпиадном движении сложилась практика, по которой программирование понимается в широком смысле. Перед участником олимпиады по программированию ставится гораздо более широкая задача, чем просто записать алгоритм на языке программирования. Требуется по условию задачи, данному в свободной форме, формализовать задачу, построив информационную и математическую модель, разработать алгоритм решения задачи, реализовать алгоритм на наиболее подходящем языке программирования (из числа разрешенных), протестировать решение, отправить на проверку автоматизированной системе. Чтобы справиться с этой задачей, от участника требуются знания алгоритмов, сетевых технологий, операционных систем, умения работы с многими приложениями — все то, что относят к области информатики. Поэтому олимпиады по информатике и по программированию суть одно и то же, выбор названия зависит от традиций.

Олимпиады по программированию/информатике можно рассматривать как интеллектуальные состязания молодежной программистской элиты. Подготовка участников и формирование команд опирается на образовательную, научную, культурную компьютерную инфраструктуру учебного заведения. Результаты, показываемые участниками на олимпиадах по программированию, довольно точно отражают состояние компьютерной инфраструктуры в учебном заведении<sup>3</sup>, а в случае Международной студенческой олимпиады по программированию (ICPC) — в различных странах. Математико-программистское содержание олимпиад по программированию понятно немногим специалистам и педагогам высшей квалификации. Однако результаты соревнований по программированию представляют интерес уже для гораздо более широких кругов.

Для организации олимпиады жюри должно подготовить комплект задач (на нашем чемпионате обычно 5–8). Задачи должны быть оригинальными, их сложность должна соответствовать уровню участников, при этом тексты задач, примеры входных и выходных данных, тесты и чекеры (программы проверки соответствия ответа программы эталону) должны быть подготовлены качественно и не должны содержать ошибок. Чемпионат в СГУ проводится обычно в 3 тура, а это значит, что ежегодно готовятся пакеты примерно для 20 задач. Приведем пример

---

<sup>3</sup>Так, в СГУ на командные туры регистрируется 15–18 команд, это соответствует 45–54 участникам. На личном туре 2017 года было 75 человек. Все участники обеспечиваются однотипными компьютерами с однотипным программным обеспечением.

олимпиадной задачи.

**D. Теорема Пика (командный тур чемпионата СГУ, 2018 г.)<sup>4</sup>**

Авторы: Н.О. Котелина, Н.К. Попова

Имя входного файла: стандартный ввод

Имя выходного файла: стандартный вывод

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Учитель математики Александра — очень образованный человек. Сегодня он разбирает с Александром соотношение между площадью многоугольника и количеством точек с целочисленными координатами, лежащими строго внутри и на сторонах многоугольника. Это нетривиальное соотношение открыл австрийский математик Георг Александр Пик в 1899 году. Пусть на плоскости с декартовой системой координат дан некоторый многоугольник с ненулевой площадью  $S$ . Обозначим количество точек с целочисленными координатами, лежащих строго внутри многоугольника через  $I$ , а количество точек с целочисленными координатами, лежащих на сторонах многоугольника, через  $B$ . Тогда  $S = I + B/2 - 1$ . Александру требуется по известной площади многоугольника и координатам его вершин вычислить, сколько точек с целочисленными координатами лежит строго внутри многоугольника.

**Формат входных данных**

В первой строке задано число  $N$  ( $3 \leq N \leq 1000$ ) — количество вершин многоугольника, во второй строке вещественное число  $S$  — площадь многоугольника. Далее следует  $N$  строк, содержащих целочисленные координаты вершин многоугольника  $x_i, y_i$ , не превосходящие по модулю  $10^6$ , в порядке обхода по или против часовой стрелки (последовательно, одна за одной).

**Формат выходных данных**

Количество точек с целочисленными координатами, лежащих строго внутри многоугольника.

---

<sup>4</sup>Подобную, но более сложную задачу можно найти по ссылке: <http://informatics.mccme.ru/mod/statements/view3.php?id=1160&chapterid=456#1>.

**Пример**

Стандартный ввод	Стандартный вывод
4	1
4	
-1 -1	
-1 1	
1 1	
1 -1	

**Этап формализации.** Задача имеет математическое содержание, требует «догадки», но и дает подсказку. Условие задачи надо дочитать до конца. Только в описании формата входных данных находится важная для решения задачи информация: координаты вершин многоугольника — целые числа. Если участник нарисует достаточно много многоугольников с вершинами с целочисленными координатами, он может заметить, что стороны многоугольника (отрезки прямых) могут быть вертикальными или горизонтальными. Тогда, чтобы узнать количество целых точек на отрезке, надо вычесть координаты концов и добавить единицу. Интерес представляет случай, когда отрезок не является вертикальным или горизонтальным. В этом случае можно заметить, что если рассматривать отрезок как гипотенузу прямоугольного треугольника, то ответом будет число, равное наибольшему общему делителю длин катетов этого треугольника плюс единица. Для стороны с концами в точках  $(x_1, y_1)$ ,  $(x_2, y_2)$  это количество равно  $\text{НОД}(|x_1 - x_2|, |y_1 - y_2|) + 1$ . Доказывать этот факт от участника не требуется. Количество целых точек на границе многоугольника получится, если сложить количества целых точек для всех сторон многоугольника. Но при сложении каждая концевая точка будет подсчитана дважды, поэтому количество целых точек должно быть уменьшено на  $N$  — количество вершин. Осталось применить теорему Пика и вывести  $I = S - B / 2 + 1$ .

Для формализации задачи от участника требуется хорошая математическая подготовка, наблюдательность и развитая интуиция. Итак, задача сведена к выделению частных случаев и нахождению наибольшего общего знаменателя двух натуральных чисел в общем случае.

**Построение алгоритма, программирование и тестирование**

Для нахождения наибольшего общего делителя разумно использовать алгоритм Евклида. Его следует аккуратно применять, поскольку координаты вершин — целые числа, а входные данные для правильной работы алгоритма Евклида — натуральные числа. Для написания программы можно использовать любой универсальный язык програм-

мирования. С помощью правильно подобранных тестов можно будет аккуратно проверить частные случаи. Сложность задачи оценивалась составителями как средняя. Назначение задачи — отделить от основной массы участников тех, кто будет бороться за призовые места<sup>5</sup>.

На командном туре Открытого чемпионата СГУ по программированию 2017 года было 7 задач. Для каждой из них жюри проработало формулировку задачи, авторские решения, систему тестов и чекеры. В идеале подготовкой комплектов олимпиадных задач должен заниматься программный комитет. В его обязанности входит поддержка банка задач, проработка идей задач и составление строгих формулировок, формирование комплекта задач для конкретных соревнований, а также непосредственно подготовка задач: составление условий, разработка системы тестов, авторских решений, программ проверки корректности входных данных (валидаторов), программ проверки ответов участников (чекеров). Например, в УрФУ<sup>6</sup> программный комитет состоит из 10–15 человек: старшекурсники, аспиранты и выпускники, которые в прошлом активно участвовали в соревнованиях по программированию.

В нашем университете все гораздо скромнее. Подготовкой задач занимается 2–3 человека, сейчас — преподаватели кафедры, в прошлые годы — преподаватели, студенты-старшекурсники и выпускники. Банк задач (идей задач) формируется эпизодически, в основном используются уже известные задачи, которым дается новая интерпретация. Причина этому — нехватка квалифицированных специалистов. Работа в программном комитете и жюри ведется на некоммерческой основе, в свободное от основной работы время. Однако энтузиазм не основа для долговечной структуры, поэтому со временем студенты и выпускники теряют желание или возможность тратить свое личное время на подготовку задач.

Проверку правильности программ осуществляет автоматизированная система проверки решений (электронный судья). Программы тестируются на большом количестве различных входных тестов, неизвестных участникам. Если программа выдала неправильный ответ или не уложилась в ограничения по времени или памяти, то пославшая её команда получает об этом сообщение и может послать исправленную версию. Задача считается решённой, если программа выдала правильные ответы на всех тестах. Частичные решения не учитываются. Этот

---

<sup>5</sup>См. протокол чемпионата. URL: <https://nkropova.blogspot.com/2018/04/xix-2018.html>.

<sup>6</sup>Самсонов А., Ипатов А. Система подготовки олимпиадных задач по программированию в УрГУ. URL: <http://acm.urfu.ru/library/system.html>

формат предъявляет определенные требования к системе тестов и способу проверки соответствия тестам. Другими требованиями к проверяющей системе являются стабильность работы при большом количестве участников, достаточная вычислительная мощность тестирующего сервера для быстрой проверки решений, чтобы не возникало очередей проверок, простота использования, возможность получить справочную информацию и связаться с группой поддержки по поводу возникающих проблем.

С 1999 года неоднократно менялась платформа проверки решений. Отсутствие в первые годы хороших проверяющих систем привело к необходимости разработки в 2002 году собственного контестера «ОКО», он хоть и имел ограниченный функционал, но был полностью прозрачен в плане настройки, а также давал возможность взаимодействия пользователей и жюри соревнования. Далее ему на смену пришла система тестирования Contester, установленная также на сервере университета. К числу бесспорных достоинств этой системы относится простота установки и использования. К сожалению, эта система не выдерживает больших нагрузок и недостаточно гибкая, например, не в полной мере поддерживает библиотеку `testlib`, из-за чего при проведении олимпиад на ее базе приходилось переписывать проверяющие программы. Не поддерживает никакие форматы соревнований, кроме АСМ/ICPC, и теперь уже не поддерживается разработчиком. Поэтому начиная с 2017 года организаторами чемпионата Н. О. Котелиной и Н. К. Поповой было принято решение перейти на платформу Яндекс.Контест. В первом же из чемпионатов на базе данной платформы приняли участие 15 команд. Платформа Яндекс.Контест удовлетворяет практически всем сформулированным выше требованиям и может использоваться для разных целей.

#### **4. О влиянии спорта на учебный процесс**

В первом городском открытом чемпионате по программированию (см. [2]) в марте 2000 года участвовали студенты СГУ, пединститута, школьники КРФМЛИ. Соревнование выиграла команда КРФМЛИ (Спеськов Александр, Белоголов Станислав, Думина Юлия). Е. И. Михайловский, заведующий кафедрой математического моделирования и кибернетики, был недоволен и сделал оргвыводы. В этом же году был организован факультатив по решению олимпиадных задач по программированию для студентов 4-го курса математического факультета. Его проводили преподаватели кафедры прикладной математики П. А. Голвач, В. Е. Езовских, А. Б. Певный, Н. К. Попова.

В 2001 году на математический факультет поступили Е. Ключников,



А. Спеськов Несколько лет они успешно выступали в студенческом чемпионате мира (ICPC), а также в рамках научно-исследовательской работы под руководством старшего преподавателя кафедры прикладной математики В. Е. Езовских спроектировали и реализовали автоматизированную систему проверки решений ОКО<sup>7</sup>. Система создает среду для автоматической проверки и мониторинга присылаемых решений. Разработанная система ОКО сразу начала активно использоваться в учебном процессе. С тех пор чемпионат по программированию дал немало тем для студенческих разработок.

В процессе обучения программированию на практических занятиях студенты изучают алгоритмы, структуры данных, пишут учебные программы. Доказательство правильности программы [3] можно и нужно использовать в основном во время разработки алгоритма решения задачи. Завершающим же этапом программирования является тестирование программы, то есть выявление особенностей представленного решения на наборе специально подобранных тестов. Для студентов отладка и тестирование собственных программ оказывается сложной. Идея поиска ошибок в программе с трудом воспринимается составителем этой программы. Подсознательно автор программы выбирает тестовые примеры, демонстрирующие работоспособность программы.

С другой стороны, проверка преподавателем правильности выполнения учебных заданий, является наиболее трудоемкой частью процесса обучения. Ввод начальных данных, хотя бы для демонстрации ошибочности программы, требует времени и внимательности. Студент значительную часть времени на занятии тратит на ожидание проверки программы преподавателем.

Тестирование уже много лет используется как средство оценки знаний и умений в разных областях. Тестирование программы является общепринятым способом при оценке правильности решения задачи на соревнованиях по программированию. Составить качественную, полную систему тестов для проверки учебной или олимпиадной задачи по программированию непросто. Зато такая система тестов зачастую может оказаться для преподавателя гораздо более полезной, чем попытка найти все возможные ошибки или же, наоборот, доказать правильность уже существующей программы с помощью анализа листинга предложенного решения. Если программа успешно прошла все тестовые испытания, преподавателю имеет смысл просмотреть ее листинг вместе со студентом, отметить недостатки (если таковые имеются) в эффективности

---

<sup>7</sup>Эта система была зарегистрирована в Информационно-библиотечном фонде РФ (№ОФАП 2711 №г.р. 50200300545).

программной реализации алгоритма и в стиле программирования.

Используя систему тестов и автоматизированную систему проверки решений, преподаватель получает больше возможностей для аналитической работы с программами студентов, в частности для проверки их эффективности по времени и объему используемой памяти.

Применение автоматизированной системы на занятиях требует от преподавателя предварительно проделанной дополнительной работы. В частности, требуется сформулировать условие задачи так, чтобы оно не допускало разночтений и понималось однозначно. Например, неправильная формулировка: «Написать программу сортировки числового массива по неубыванию».

Для представления в автоматизированной системе надо конкретизировать условие. Например, так. Написать программу сортировки массива, состоящего из целых чисел, каждое из которых по модулю не превосходит 32000, по неубыванию. Входные данные вводятся с клавиатуры. В первой строке ввода находится число  $N$  — размер массива, в следующих строках —  $N$  элементов массива, разделенных пробелами и/или символами перевода строки. Результат работы программы —  $N$  отсортированных чисел, вывести в текстовый файл OUTPUT.TXT, разделяя числа пробелами и/или символами перевода строки. Время работы программы на одном тесте не должно превышать 3 секунд.

**Example.**

Input	Output
3	-1
4 -1	2
2	4

Таким образом, постановка задачи (спецификация задания) должна обладать свойствами четкости и полноты. Кроме того, желательно давать неформальную постановку задачи с тем, чтобы студент выполнил самостоятельно этап формализации задачи.

Вот пример типичной учебной задачи по теме «Массивы и строки». Сначала дается содержательная формулировка, возможно, раскрывается смысл некоторых понятий.

**Зачёт по сетям**

Идёт зачёт по курсу «Компьютерные сети». Надо по IP-адресу компьютера определить, удовлетворяет ли этот адрес имеющейся маске. Соответственно, Вам дан адрес в десятичной нотации с точками и битовая маска. Ваша задача — написать программу проверки, имеется ли в адресе последовательность из подряд идущих битов, совпадающая с маской.

Адрес каждой машины в Internet представляет собой строку из 32 бит, состоящую из двух частей. Часть адреса, определяющая домен, называется сетевым идентификатором. Оставшаяся часть адреса, определяющая конкретную машину в домене, называется адресом узла. Адреса традиционно записывают в десятичной нотации с точками. Это означает, что 32 бита разбивают на 4 байта, каждый байт записывают десятичным числом и отделяют один от другого точками. Например, адрес 1000100000010000000001111100011 в десятичной нотации с точками будет записан как 136.16.7.227.

**Input.** Две строки. В первой — 4 десятичных числа, разделенных точками. Гарантируется, что даны числа в диапазоне байта. Вторая строка состоит из символов 0 и 1.

**Output.** Одна строка, содержащая YES, если в двоичном представлении адреса имеется маска, NO – в противном случае.

**Example.**

Input	Output
136.16.7.227 00000000	YES

Для решения задачи требуется преобразовать IP-адрес в строку из нулей и единиц. Например, прочитать IP-адрес как строку, выделить числа, получить их двоичное представление, выполнить конкатенацию и проверить вхождение заданной битовой строки. Хотя для её решения надо выполнить несколько процедур, каждая из них достаточно проста. Её мера трудности в системе 80 %, то есть подавляющее число студентов справились с задачей.

Другая задача по теме «Типы данных» выглядит гораздо проще.

**Золотое сечение**

Даны два числа на числовой прямой. Рассматривая их как концы отрезка, получить точку золотого сечения. *Золотым сечением отрезка называется точка, которая делит отрезок так, что его большая часть так относится к его меньшей части, как длина самого отрезка к его большей части.*

**Input:** одна строка, содержащая два числа, напечатанные через пробел.

**Output:** одна строка, содержащая золотое сечение, округленное до четырех цифр после десятичной точки.

**Example:**

Input	Output
0 1	0.6180

Для решения задачи надо составить и решить квадратное уравне-

ние, однако трудность задачи составляет 9 %. Это связано как с тем, что студенты затрудняются в составлении уравнения (в построении математической модели), так и с тем, что в задаче используются действительные числа и возникает проблема с точностью вычислений.

Для каждой задачи преподаватель разрабатывает систему тестов. Конечно, не для всякой программы нужен полный набор тестов, и преподаватель сам решает, какие качества программы особенно нуждаются в проверке.

Ещё раз повторим, программа, прошедшая все тесты, необязательно является правильной<sup>8</sup>. Требуется просмотр её листинга и обсуждение преподавателем вместе с исполнителем.

Важно отметить, что выполненная один раз работа по формулировке задач и построению системы тестов сохраняется и может быть использована многократно. Студенты получают возможность решать задачи в любое время (не только на занятиях) и проверять решение в автоматизированной системе, тем самым обеспечивается возможность для самостоятельной работы студентов.

Использование автоматизированной системы преподавателем программирования делает труд преподавателя более квалифицированным, повышает эффективность и интенсивность занятий в несколько раз, позволяет увеличить интерес студентов к данному предмету. Действительно, на занятиях появляется игровой момент, студенты вовлекаются в соревнование, с другой стороны, каждый работает в своем темпе и не связан ожиданием одобрения преподавателя.

Таким образом, использование автоматизированной системы для контроля текущей успеваемости позволяет получить количественную оценку, не зависящую от степени внимательности преподавателя и наличия у него достаточного времени для полного тестирования программы.

### **5. О тех, кто сделал значительный вклад в развитие чемпионата по программированию в СГУ**

В проведении Открытого чемпионата СГУ в большей или меньшей степени принимали участие почти все члены кафедры прикладной математики.

Душой и организатором Открытого чемпионата СГУ по программированию был заведующий кафедрой В. Л. Никитенков. Он взял на себя всю бюрократическую волокиту, связанную с оформлением различной документации, разработал положение и правила проведения чемпиона-

---

<sup>8</sup>Если программа прошла все тесты, то система тестов не полна.

та. Неоднократно выезжал с командами университета на чемпионаты мира в качестве сопровождающего тренера. Предложенные им сертификаты участника до последнего времени получали студенты и школьники, принявшие участие в мероприятии. При Владимире Леонидовиче спонсором чемпионата была фирма «Содействие». Участники чемпионата тех лет до сих пор помнят о «полдниках» — легком перекусе, организуемом с помощью «Содействия», поскольку чемпионат длился пять часов. А введенные Владимиром Леонидовичем призы — ручки Parker — и сейчас служат наградой лучшим.

Огромный вклад в проведение чемпионатов сделал старший преподаватель кафедры прикладной математики В. Е. Езовских. Владимир Евгеньевич создавал оригинальные задачи с условиями на английском языке, пакеты тестов. Он обладал удивительным талантом придумывать тему для очередной задачи буквально на ходу и всегда подходил к этому процессу творчески и с огромным азартом, поэтому неудивительно, что увлек спортивным программированием многих студентов. Был тренером команд кафедры при подготовке к чемпионатам мира по программированию, в том числе тренером команды DWED. Именно им принадлежит наивысшее достижение наших студентов — диплом третьей степени на полуфинале 2004 года. Команду DWED составили Ключников Евгений, Спеськов Александр и Попов Дмитрий, запасным в команде был Шеринга Александр.

Доцент кафедры прикладной математики Н. К. Попова стояла у истоков Открытого чемпионата и является одним из ключевых организаторов, а начиная с 2015 года в число организаторов также вошла доцент кафедры Н. О. Котелина. В текущем году Открытый чемпионат СГУ им. Питирима Сорокина (далее чемпионат СГУ) по программированию вошел в межкафедральный проект «Развитие сетевого взаимодействия в области математики, физики, информатики и робототехники между образовательными организациями финно-угорских республик Российской Федерации».

С самого начала в организации и проведении чемпионатов активно участвовали студенты. Многие участники чемпионатов, каждый в свое время, побывали «по другую сторону баррикад», отметившись в числе организаторов.

Так, участник полуфинала командного чемпионата мира по программированию 1999 года Роман Зеянин входил в состав жюри чемпионата университета в первые годы. В 2002 году студентами кафедры прикладной математики Ключниковым Евгением и Спеськовым Александром была разработана проверяющая система «ОКО», активное ис-

пользование которой в соревновательном и учебном процессе было начато в 2004 году. Эстафету подхватил студент кафедры прикладной математики Максименко Владимир, а потом и Дурягин Александр. Участие в чемпионатах дало им возможность оценить свои навыки в программировании в условиях работы в команде. Поездки на чемпионат АСМ на Урал были частью студенческой жизни. Позднее Максименко Владимир активно занимался организацией и проведением олимпиад в КРФМЛИ<sup>9</sup>. Дурягин Александр работает учителем информатики в КРФМЛИ. Он высоко оценивает пользу чемпионатов для школьников. Это не только опыт, который необходим участникам олимпиад самого разного уровня и который нельзя приобрести без участия в очных турнирах — это еще и общение со сверстниками, имеющими такие же интересы, и новые знакомства, и новые друзья. Здесь могут проявить себя участники, которым сложно пробиться на региональные и всероссийские олимпиады.

Долгое время автором задач чемпионата и его администратором был аспирант кафедры Кораблев Анатолий. Его задачи отличались хитростью формулировок и необходимостью при решении учитывать неочевидные частные случаи. В марте 2013 года он реализовал идею «тряхнуть стариной» и поучаствовать в самом престижном — командном туре чемпионата университета по программированию<sup>10</sup>. С тех пор почти в каждом командном туре чемпионата принимает участие команда Old School. В разные годы в ее составе кроме Анатолия выступали Максименко Владимир, Сластухин Иван, Константин Кузнецов, Потолицын Александр, но всегда неизменно команда была в числе призеров и победителей.

Студент кафедры прикладной математики Болгар Василий, выпускник КРФМЛИ, увлекся программированием еще в 7 классе. Успешно выступал в чемпионате, команда AdorableJoke (в составе Болгар Василий, Гацковка Антон, Рябов Евгений) несколько раз участвовала в четвертьфинале студенческого чемпионата мира. Несколько лет Василий администрировал чемпионаты СГУ и составлял задачи.

Студентка кафедры прикладной математики Лукина Татьяна провела несколько чемпионатов и других соревнований на базе Контестера.

Можно много говорить об организационных моментах — сертификаты участников, призы, административное (приказы, выделение аудиторий) и информационное сопровождение — во всем этом нам помога-

---

<sup>9</sup>Об этих олимпиадах и задачах можно почитать в его блоге: <http://xupyr.com/category/it/contests/>.

<sup>10</sup><https://blog.akorablev.ru/after-contest.html>.

ют службы университета. Особенно хочется отметить безукоризненную работу сотрудников отдела управления информатизации по подготовке компьютеров и программного обеспечения к проведению соревнований по программированию.

Многое для развития чемпионата сделали спонсоры. Компания «Содействие» — лидер на региональном рынке компьютерной техники и IT-услуг, поддержала чемпионат в первые годы становления. Фирма «Цифроград» и ее глава С. В. Артеев в трудное время оказала значительную помощь в проведении чемпионатов и поездок победителей на четвертьфинал студенческого чемпионата мира. Сергей Вячеславович и сейчас как депутат и председатель Комитета Государственного Совета Республики Коми по социальной политике поддерживает нас и каждый год выступает перед участниками командного тура чемпионата СГУ.

## Список литературы

1. **Котелина Н. О., Попова Н. К.** Подготовка интернет-тура чемпионата по программированию на Yandex.Contest // *Вестник Сыктывкарского университета. Сер. 1: Математика. Механика. Информатика. 2018. Вып. 1 (26). С. 73–79.*
2. Комиинформ. <https://komiinform.ru>. Первый городской открытый чемпионат по программированию прошел в Сыктывкаре в выходные. URL: <https://komiinform.ru/news/4351> (дата обращения: 12.12.2018).
3. **Кирюхин В. М.** Методика проведения и подготовки к участию в олимпиадах по информатике. Всероссийская олимпиада школьников. М.: БИНОМ. Лаборатория знаний, 2011. 271 с.

### Summary

**Kotelina N. O., Popova N. K., Yurkina M. N.** About open championship of SSU on programming

The article is devoted to the jubilee, XV Open Syktyvkar State University Programming Championship. It tells about the experience of the event, as well as about the people who have made a significant contribution to the olympiad movement.

*Keywords: sports programming, ACM, ICPC.*

## References

1. **Kotelina N. O., Popova N. K.** Podgotovka internet-tur chempionata po programmirovaniyu na Yandex.Contest (The preparation of the online round of the championship on programming on Yandex.Contest platform), *Bulletin of Syktyvkar University, Series 1: Mathematics. Mechanics. Informatics*, 2018, 1 (26), pp. 73–79.
2. Komiinform. <https://komiinform.ru>. Pervyy gorodskoy otkrytyy chempionat po programmirovaniyu proshel v Syktyvkare v vykhodnyye (The first city open programming championship was held in Syktyvkar at the weekend). <https://komiinform.ru/news/4351> (the date of circulation: 12.12.2018).
3. **Kiryukhin V. M.** *Metodika provedeniya i podgotovki k uchastiyu v olimpiadakh po informatike. Vserossiyskaya olimpiada shkol'nikov* (Methods of carrying out and preparing for participation in computer science competitions. All-Russian School Olympiad). M.: BINOM. Laboratory of Knowledge, 2011, 271 p.

**Для цитирования:** Котелина Н. О., Попова Н. К., Юркина М. Н. Об открытом чемпионате СГУ по программированию // *Вестник Сыктывкарского университета. Сер. 1: Математика. Механика. Информатика*. 2018. Вып. 3 (28). С. 3–18.

**For citation:** Kotelina N. O., Popova N. K., Yurkina M. N. About open championship of SSU on programming, *Bulletin of Syktyvkar University. Series 1: Mathematics. Mechanics. Informatics*, 2018, 3 (28), pp. 3–18.

СГУ им. Питирима Сорокина

Поступила 12.12.2018