

## АВТОМАТИЗАЦИЯ ПРОВЕДЕНИЯ ДИСТАНЦИОННЫХ СОРЕВНОВАНИЙ, ОСНОВАННЫХ НА ИССЛЕДОВАТЕЛЬСКИХ СЮЖЕТАХ ПО МАТЕМАТИКЕ И ИНФОРМАТИКЕ

Атхит Майтараттанакон,  
Посов Илья Александрович

### Аннотация

В статье описаны технологии, на базе которых проводится дистанционный исследовательский конкурс КИО (Конструируй, Исследуй, Оптимизируй), и подробно описан процесс подведения итогов, реализующий модель оценки учебной деятельности учащихся по результатам решения задач. Для описания процесса сравнения решений описывается последовательность параметров вместе с указанием метода сравнения параметров. Описание дополнительно задает способ отображения значений параметров для жюри и участников.

**Ключевые слова:** исследовательские соревнования, дистанционные соревнования, оценка решений, таблица результатов.

### ВВЕДЕНИЕ

Сейчас все более популярными становятся дистанционные соревнования и конкурсы. Участвовать в таких конкурсах можно в комфортной обстановке у себя дома или в школе, а организаторам требуется один раз автоматизировать процесс, после чего их основной обязанностью остается только подготовка заданий и, в случае творческих конкурсов, оценка работ участников.

Ранее дистанционные соревнования ассоциировались прежде всего с олимпиадами по программированию. В них проверка решения неизбежно производится автоматически, поэтому они естественным образом превращаются в удаленные. Одним из примеров организаций, регулярно проводящих удаленные соревнования по программированию, является TopCoder [1]. Сайт Timus Online Judge [2] имеет архив задач по программированию и позволяет участвовать в онлайн версиях большинства соревнований, проводящихся в Уральском федеральном университете.

Сайт Project Euler (Проект «Эйлер») [3] предлагает решать задачи по математике, и, хотя в явном виде соревнования на нем не проводятся, пользователи соревнуются в том, кто решит новые появляющиеся задания быстрее.

Приведенные выше примеры удаленных соревнований рассчитаны на высокий уровень подготовки участников, но удаленные соревнования и конкурсы существуют для широкого круга участников. Поисковый запрос «удаленные конкурсы» выдает внушительный список сайтов, которые регулярно проводят соревнования и конкурсы по различным дисциплинам для всех желающих.

Одним из особенностей удаленного проведения соревнования является то, что оно легко масштабируется по количеству участников, особенно в случае автоматизированной проверки решений. Возможность автоматизации проверки зависит от типов задаваемых вопросов, поэтому актуальной является задача автоматизации проверки для как можно более разнообразных типов вопросов. В этой статье рассматривается задача автоматизации проверки исследовательских заданий. В исследовательских соревнованиях победителями являются те участники, которые дальше продвинулись в исследовании некоторой проблемы и получили результаты, лучше, чем у других участников.

Мы будем рассматривать автоматизацию проверки решений в заданиях конкурса «КИО» [4–8], кроме того, мы рассмотрим автоматизацию всего процесса проведения соревнования.

## ОБЩАЯ КАРТИНА

Задания конкурса «КИО» — это исследовательские задания, в которых участнику предоставляется виртуальная лаборатория для проведения экспериментов. Участник предлагает решения, лаборатория автоматически их проверяет, запоминает лучшее. При подведении итогов строится таблица результатов на основе того, кто из участников продвинулся в исследовании дальше.

Задания должны отвечать ряду требований, основные из них таковы: задание является исследовательским, это означает, что оно предполагает оптимизацию, при этом жюри часто не знает оптимального решения задачи, например, потому что оно основано на одной из нерешенных проблем. Кроме того, все решения заданий должны подробно оцениваться, чтобы ни один участник не получил нулевой оценки, и решения всех участников можно было бы сравнить друг с другом. Приведем примеры условий заданий разных лет, демонстрирующие процесс оценки решений. Это всего 4 примера из порядка 30 лабораторий, они выбраны по принципу наибольшего разнообразия параметров, на основе которых сравниваются решения.

Задача Томсона [9]: участники располагают на сфере несколько электрических зарядов, их задача сделать так, чтобы общая энергия системы была как можно меньше.

Задача «Качели» [10] предлагает написать программу для робота, раскачивающегося на качелях, чтобы он как можно быстрее сделал полный оборот вокруг оси. При оценке решения сначала сравнивается время, которое потребовалось роботу для оборота (время округляется до секунд). Если у двух участников совпадает время, то решения сравниваются по длине программы — она должна быть как можно короче.

В задаче «Поезда» требуется катать вагоны поезда по рельсам так, чтобы они встали в правильном порядке. Авторы задачи предполагали сравнивать участников только по количеству действий, необходимых для упорядочивания. Но в результате было создано четыре параметра: сначала сравниваются, сколько вагонов стоят на нужных рельсах, если эти значения совпадают, сравнивается количество транспозиций в перестановке вагонов, их должно быть как можно меньше. Если и количество транспозиций совпадает, сравнивается количество операций по поднятию вагонов в горку. Если и эти значения совпадают, сравнивается общее количество операций.

Последний пример — задача «Паук». В ней нужно собрать шагающего паука. Участники сравниваются по тому, чей паук дошел до финиша быстрее, время округляется до се-

кунд. Если время совпадает, то сравнивается количество использованного материала, оно должно быть как можно меньше.

## ТЕХНОЛОГИИ

Программное обеспечение конкурса состоит из трех частей:

1. Система проведения соревнования
2. Программная оболочка конкурса
3. Модуль проверки решений

Система проведения соревнований отвечает за организационную составляющую соревнования, она выполнена в виде Интернет-сайта, на нем участники читают правила, регистрируются, узнают новости, загружают себе задания конкурса (программную оболочку конкурса), загружают обратно свои решения, узнают результаты соревнования.

Программная оболочка конкурса содержит три лаборатории с задачами соответствующего года. Она позволяет решать задачи и сохранять решения, для того чтобы загрузить их в систему проведения соревнований для последующей проверки.

Модуль проверки решений проверяет решения участников и оценивает их. Результаты проверки решений загружаются в систему проведения соревнований, чтобы на их основе подвести итоги соревнования.

Система проведения соревнований разработана с помощью Play Framework и написана на Java, она позволяет проводить различные виды соревнований и основана на идеях, описанных в статье [11]. Кроме конкурса «КИО», на ее основе проводится конкурс «Бобер» [12, 13]. Система позволяет настраивать временные параметры соревнований, то есть кто, когда и при каких условиях получает доступ к условиям заданиям, результатам проверки своих заданий и к общим результатам соревнования. Система собирает решения участников и позволяет гибко управлять процессом их оценки, об этом будет написано ниже.

Программная оболочка конкурса разработана на платформе Flash с помощью языка ActionScript, на данный момент это наиболее удобная платформа для быстрой разработки приложений, интенсивно использующих графику и анимацию, и в недавнем времени она была доступна на всех популярных платформах, включая мобильные. К сожалению, сейчас распространение Flash уменьшается, а приходящий на смену аналог HTML5 далеко не так удобен в разработке, и производительность его не сравнима с Flash.

Важной особенностью программной оболочки является то, что она не пользуется доступом к Интернету, участникам предлагается сохранять свои решения на диск, после чего вручную загружать их на сайт. Эта особенность оболочки появилась во времена, когда большинство участников не имели доступа в Интернет, кроме как в школе или использовали Интернет с повременной оплатой. Такие участники имеются до сих пор, поэтому оболочка и далее будет работать без Интернета, но для нее планируется возможность работать в режиме онлайн.

Оболочка позволяет сохранять решения участника в файл, такой файл содержит информацию об участнике (сейчас это только фамилия и имя) и о решении каждой из трех задач. Этот файл загружается в систему проведения соревнований. Кроме этого, оболочка позволяет сохранять решения отдельных задач. Это необходимо, для того чтобы участник не боялся изменять свои решения, сохранял их и мог вернуться к ним позже. К сожалению, две разных возможности сохранения путают участников, они пытаются отсылать для проверки вместо одного файла со всеми тремя задачами файлы с решениями отдельных задач или другие произвольные файлы. Система проведения соревнований при загрузке анализирует полученный файл и пытается выдавать осмысленные сообщения об ошибках. Проблема неправильных загрузок — это еще одна причина, по которой возможность оболочки работать онлайн является актуальной.

Модули проверки решений оценивают решения участников, что представляет собой оценка, будет указано ниже. Оценки получает система проведения соревнований, сравнивает на их основе решения участников между собой и формирует таблицу результатов, о построении таблицы написано ниже.

Модули проверки решений нужны только для того, чтобы избежать нечестных действий со стороны участников. Реально оценкой решений занимается оболочка. Она оценивает каждое решение, полученное участником, и запоминает наилучшее решение. Именно наилучшее решение сохраняется в файл со всеми решениями участника. Вместе с самими решениями можно было бы сохранить и их оценку, после чего использовать их для построения таблицы результатов, но участники теоретически могут исправить файл с решениями таким образом, чтобы улучшить оценку своих решений. Защититься от этого невозможно, потому что оболочке для этого придется выступать в роли доверенного клиента [14]. Поэтому приходится проводить повторную проверку решений в доверенной среде.

Модули проверки решений чаще всего реализованы на Flash, потому что для этого достаточно взять и так уже написанный код в оболочке. Но в последнее время проверка переносится на Java, подобное дублирование логики имеет ряд преимуществ. Оно позволяет обнаружить ошибки в реализации проверки на ранней стадии, совершать проверку внутри системы проведения соревнований, что значительно уменьшает количество ручных действий на стадии подведения итогов.

К сожалению, иногда перенос логики проверки решения на Java невозможен. Это верно для задач, в которых одним из параметров оценки решений является вещественное число со сложным алгоритмом вычисления, например, на основе моделирования физического процесса. В задаче «Качели» таким параметром является время до полного оборота, а в задаче «Паук» — время движения паука. Вычисления с вещественными числами могут быть неустойчивыми, и результат зависит от порядка действий, поэтому один и тот же алгоритм может давать разные результаты при разных реализациях.

## ОЦЕНКА РЕШЕНИЙ

В различных соревнованиях оценкой решений часто является количество баллов за задание, и, чтобы сравнить решения двух участников, достаточно сравнить два числа. В исследовательских заданиях сравнение решений происходит по нескольким параметрам. В статье [11] предложено считать, что на множестве решений введено отношение предпорядка, и наиболее естественная реализация этого отношения следующая. Каждое решение оценивается, то есть отображается в конечную последовательность параметров  $(p_1, p_2, \dots, p_n)$ . Далее две оценки с наборами параметров  $(p_1, p_2, \dots, p_n)$  и  $(q_1, q_2, \dots, q_n)$  сравниваются следующим образом. Если  $p_1 > q_1$  (или  $p_1 < q_1$ ), то первое решение лучше, иначе если  $p_1 < q_1$  (или  $p_1 > q_1$ ), то второе решение лучше, иначе если  $p_1 = q_1$ , то переходим к аналогичному сравнению параметров с индексами 2. Оценки решений окажутся одинаковыми, только если все параметры попарно равны.

Выбор знака сравнения (больше или меньше) зависит от условия задания. Например, для параметра «время» в задаче «Паук» лучше будет то решение, у которого параметр меньше, а для параметра «едут верно» в задаче «Поезда» лучше будет решение, у которого параметр больше.

В системе проведения соревнований сравнение оценок решений описывается следующим образом, продемонстрируем его на примере задачи о поездах:

```
"с+і:верно",
"t-i:транспозиций",
"uh-i:подъемов",
"dh-i:спусков"
```

Здесь **c**, **t**, **uh**, **dh** — это краткие названия параметров, под этими метками значения параметров содержатся внутри структуры данных с оценкой. Знак «+» или «-» означает, как сравнивать значения параметров. В первом случае значение должно быть как можно больше, во втором — как можно меньше. Символ **i** означает тип параметра. Тип влияет на сравнение и на отображение параметра при печати. Возможные типы **i** — целое, **d** — вещественное, **b** — логическое значение, например, сумел ли пользователь достичь определенного результата. Сравнение вещественных чисел отличается от сравнения целых, потому что происходит с определенной точностью, два числа считаются равными, если различаются на значение, меньшее наперед заданного *эпсилон*. Такое сравнение не транзитивно, поэтому вещественные параметры лучше избегать, а если это невозможно, то они часто округляются до некоторого количества знаков после запятой.

После двоеточия указывается название параметра для отображения участнику. Эти названия видны:

- 1) в оболочке — их участник видит в процессе решения;
- 2) после загрузки решения на сайт — их он видит, чтобы убедиться, что загрузил то, что собирался;
- 3) в окончательных результатах соревнования;
- 4) в сертификате участника.

Порядок параметров в описании важен, именно в этом порядке происходят сравнения.

В результате участник может увидеть следующую оценку своего решения: «верно 20. транспозиций 0. подъемов 7. спусков 48.»

Приведем еще один пример описания параметров в задаче «Паук»:

```
"t-d:время~ c", "m-d:материал~ см"
```

Здесь добавились дополнительные пометки после знака тильды. Это постфиксы для значений. Например, в результате оценка решения может выглядеть как «время 59 с. материал 136 см.».

## ПОСТРОЕНИЕ ТАБЛИЦЫ РЕЗУЛЬТАТОВ

В предыдущем разделе мы описывали оценку отдельных решений участников. Пройдемся подробнее по всему процессу построения таблицы результатов в соответствии со схемой из статьи [11].

В процессе решения одной задачи участник постоянно предлагает решения. После каждого действия внутри виртуальной лаборатории, изменяющего решение, решение оценивается. Из всех решений сохраняется то, которое имеет наилучшую оценку.

От оценки отдельных решений необходимо перейти к оценке решения задачи. В общем случае при превращении оценки решения в оценку задачи могут добавиться дополнительные параметры, например время получения наилучшего решения, это нужно, если мы хотим дополнительно определять, кто получил решение быстрее.

В КИО дополнительными параметрами являются баллы и рейтинг по задаче. Количество баллов за задачу равно количеству участников, которые решили задачу хуже. Такой способ оценки представляется наиболее объективным способом оценить решение участника по сравнению с другими решениями. Кроме того, организаторы конкурса часто сами не знают оптимального решения задачи, поэтому не могут предложить осмысленной формулы, превращающей набор параметров оценки решения в баллы.

Рейтинг вычисляется на основе баллов, каждый получает рейтинг, равный 1 + количество людей, набравших больше баллов. То есть все участники с лучшим решением получают рейтинг 1.

Дальше происходит переход от оценки каждой задачи к оценке всего соревнования. Эта оценка состоит из двух чисел — баллов и рейтинга. Количество баллов равно сумме баллов по каждой задаче, рейтинг вычисляется аналогично описанному ранее.

## ЗАКЛЮЧЕНИЕ

В статье описаны технологии, на базе которых проводится дистанционный конкурс «КИО», и подробно описан процесс подведения итогов, реализующий модель из статьи [11]. Процесс сравнения оценок может быть описан формально с помощью описания параметров решений. Это описание дополнительно задает представление параметров при отображении их для жюри и участников.

Используемые технологии в будущем потребуются заменить, в первую очередь это требующий популярность Flash. А разрозненные модули по возможности объединить, чтобы упростить разработку и уменьшить объем ручной работы по переносу данных между модулями.

## Литература

1. Дистанционные соревнования по программированию. Режим доступа: <http://www.topcoder.com> (дата обращения: 01.11.2014).
2. Архив задач по программированию с проверяющей системой. Режим доступа: <http://acm.timus.ru> (дата обращения: 01.11.2014).
3. Проект «Эйлер». Режим доступа: <https://projecteuler.net> (дата обращения: 01.11.2014).
4. Сайт конкурса «Конструируй, Исследуй, Оптимизируй». Режим доступа: <http://go.kio.spb.ru/> (дата обращения: 01.11.2014).
5. *Pozdniakov S., Posov I.* Domain specific language approach to technology-enhanced learning, The Electronic Journal of Mathematics and Technology, Volume 8 (2014), Special Issue (February), 2014. P. 149–158.
6. *Posov I., Pozdniakov S.* Implementation of Virtual Laboratories for a Scientific Distance Game-Competition for Schoolchildren, in proceedings of ICAICTE 2013, August 2013. P. 495–499.
7. *Pozdniakov S., Posov I., Akimushkin V., Maytarattanakhon A.* The Bridge from Science to School, X World Conference on Computers in Education, July 2013. Vol. 3. Torun, Poland. P. 131–132.
8. *Pozdnyakov S. N., Posov I.A., Puhkov A.V., Tsvetova I.V.* Science Popularization by Organizing Training Activities Within the Electronic Game Laboratories // International Journal of Digital Literacy and Digital Competence, 3(1). January–March 2012. P. 17–31.
9. *Андреев Н.Н., Калинин М.А.* Компьютерные фильмы о занимательных и нерешенных проблемах математики. Фильм первый. Задача Томсона // Компьютерные инструменты в образовании, 2005. № 1. С. 75–81.
10. *Посов И.А.* Задача «Качели» конкурса КИО-2008 // Компьютерные инструменты в школе, 2008. № 3. С. 19–23.
11. *Посов И.А., Рукишин С.Е.* Модель оценки учебной деятельности учащихся по результатам решения задач // Компьютерные инструменты в образовании, 2011. № 3. С. 31–41.
12. Международный конкурс по информатике «Бобер». Режим доступа: <http://bebras.org> (дата обращения: 01.11.2014).
13. Российское представительство международного конкурса по информатике «Бобер». Режим доступа: <http://bebras.ru> (дата обращения: 01.11.2014).
14. *Bruce Schneier.* The Fallacy of Trusted Client Software. Information Security Magazine, August 2000. Режим доступа: [https://www.schneier.com/essays/archives/2000/08/the\\_fallacy\\_of\\_trust.html](https://www.schneier.com/essays/archives/2000/08/the_fallacy_of_trust.html) (дата обращения: 01.11.2014).

## AUTOMATION OF DISTANCE CONTESTS BASED ON RESEARCH PROBLEMS IN MATHEMATICS AND INFORMATICS

Athit Maytarattanakhon, Posov I.A.

### Abstract

The paper describes technologies used to hold a distance research competition CTE (construct, test, explore). The process of summarization of competition results is described in details. It implements the model of learning activity assessment based on problem solving results. To describe the process of solution comparison the sequence of parameters is described together with the method of parameter comparison. This description also provides a way to display parameter values for the jury and the participants.

**Keywords:** *research contests, online contests, solution assessment, results table.*

Атхит Майтараттанакон,  
аспирант СПбГЭТУ «ЛЭТИ»,  
учитель в школе Амнуайсилпа,  
(Бангкок, Тайланд),

Посов Илья Александрович,  
ассистент кафедры ВМ-2  
СПбГЭТУ «ЛЭТИ»,  
старший преподаватель кафедры  
информационных систем в искусстве  
и гуманитарных науках СПбГУ,  
iposov@gmail.com.



Наши авторы, 2014.

Our authors, 2014.