

## УПРАВЛЕНИЕ ПРОЦЕССАМИ ПРОВЕРКИ РЕШЕНИЙ И ПОДВЕДЕНИЯ ИТОГОВ В СОРЕВНОВАНИЯХ И ТЕСТИРОВАНИЯХ С АВТОМАТИЗИРОВАННОЙ ПРОВЕРКОЙ РЕШЕНИЙ

### Аннотация

Статья посвящена описанию процесса оценки результатов решений задач в системе DCES (Distance Competitive Educational System), позволяющей проводить удаленные соревнования и тестирования с автоматизированной проверкой решений задач и автоматическим подведением итогов соревнований. Оценка решений участников происходит в четыре этапа: объективная оценка решения, перенос информации в таблицу результатов, вычисление окончательных результатов, сортировка участников. Для каждого этапа описаны происходящие процессы и возможности по их настройке.

**Ключевые слова:** Образование, Система проведения удаленных соревнований, автоматическое тестирование, подведение итогов.

### ВВЕДЕНИЕ

В этой статье мы обсудим управление процессами обработки решений задач в системе DCES (Distance Competitive Educational System), позволяющей проводить удаленные соревнования и тестирования с автоматизированной проверкой решений задач и автоматическим подведением итогов соревнований. Термин DCES для такого рода систем впервые, по-видимому, введен в статье [1]. Несмотря на то, что система в первую очередь предназначена для решения задач по математике, информатике и другим точным и естественным наукам, конкретный учебный предмет не имеет особого значения: она позволяет проводить любые учебные и состязательные мероприятия, в которых

участники решают задачи и сдают их решения для автоматической или полуавтоматической проверки. Таким образом, кроме учебных тестирований, экзаменов и олимпиад, ее можно использовать, например, для организации учебного процесса в ситуации, когда преподаватель в течение учебной четверти или семестра собирает решения задач школьников или студентов и для каждого из обучаемых подводит итоги его деятельности в зависимости от количества и сложности решенных им задач, определяя итоговый рейтинг каждого студента или школьника. Далее, за неимением лучшего термина, в этой статье мы будем говорить о *соревнованиях*, имея в виду любые учебные и конкурсные формы работы, связанные с процессами сбора решений задач, их проверки и упорядочения (полного или

частичного) участников в зависимости от достигнутых ими результатов.

Далее мы обсудим возможности системы DCES и подробнее остановимся на управлении процессом определения результатов соревнований. Самими программами для проведения тестирований сейчас трудно кого-либо удивить, они встроены во все системы управления обучением. Самыми известными из них являются Moodle, Sakai, LRS. Дополнительно к ним, в Интернете имеется ряд платных и бесплатных сервисов, позволяющих провести онлайн-тестирование. Одним из такого рода примеров является сервис «Google Документы», который позволяет создать и раздать тестируемому форму со списком вопросов. Ответы участников собираются в excel-подобную таблицу (таблица google-документов), с которой можно произвести любые вычисления и произвольную обработку результатов. Помимо этого, существуют отдельные специализированные программы для проведения определенных фиксированных типов соревнований. Так, например, существует ряд проверяющих систем для проведения олимпиад по программированию, в которых участники посылают исходные тексты своих программ на автоматическую компиляцию и проверку на заранее заданных тестах. В отличие от них, система DCES не является узкоспециализированной и имеет универсальный характер.

Система DCES реализована как распределенное приложение и состоит из сервера соревнований и модулей участников [2]. Участники решают задачи и посылают их на проверку, сервер соревнований хранит информацию о проходящих соревнованиях, получает и проверяет решения участников. Современные тенденции таковы, что все больше приложений переходит в Интернет, то есть имеют веб-интерфейс и, таким образом, требуют для работы только браузер без установки дополнительных программ. Тем не менее, модуль участника в системе DCES решено реализовать как отдельную Java – про-

грамму. Причины такого выбора приведены ниже.

Система DCES универсальна также с точки зрения поддержки широкого класса форматов и регламентов соревнований. Согласно принятой классификации научных соревнований с автоматизированной обработкой решений [3], соревнования могут различаться по следующим основным классификационным признакам:

- методы ввода решений;
- способы отображения условий задач;
- правила отсылки решений;
- алгоритмы проверки решений;
- режимы доступа к результатам соревнования;
- процесс подведения итогов.

Система имеет возможности настройки всех перечисленных особенностей во всех допустимых сочетаниях, порождающих реализуемый тип соревнований. Обсудим их подробнее и сравним с возможностями других аналогичных продуктов. Одной из основных особенностей системы является поддержка разнообразных способов ввода решений. Другие системы проведения тестирований обычно предлагают ограниченный набор методов ввода. Чаще всего участник может выбирать один вариант ответа из нескольких предложенных (так называемые тесты multiple choice), вводить строку текста, которая дальше будет сравнена с правильным ответом, вводить ответ в форме эссе, которое будет прочитано и оценено преподавателем, как это делается при проверке части С ЕГЭ. Но для математических задач существуют и более естественные способы ввода. Так, скажем, в геометрической тематике естественнее предложить участнику динамический чертеж, с помощью которого он будет решать задачу или вводить ответ. Например, в задачах на построение ответом является чертеж, на котором нарисованы необходимые объекты. Другой, более характерный для математических задач случай – когда ответ вводится в виде формулы. Сложные формулы удобно вводить с помощью специ-

альных инструментов наподобие Microsoft Equation, либо же вводить с клавиатуры в нотации TeX, что вызывает необходимость просматривать получающиеся формулы по мере ввода. Система DCES позволяет расширять способы ввода решений посредством подключения, помимо этих стандартных, других дополнительных модулей. Модули имеют открытую спецификацию и могут создаваться сторонними разработчиками. Сама философия развития системы такова, что она должна распространяться с максимальным количеством уже созданных модулей, чтобы необходимость писать собственный модуль ввода у пользователей возникала как можно реже, а в случае подключения новых модулей их использование должно происходить без изменения архитектуры системы. Сказанное верно и в отношении других модулей, не связанных со вводом условий.

Особенностью системы является то, что модуль участника реализован в виде Java – приложения, а не веб–приложения. Причиной такого выбора стала относительная простота программирования сложных интерфейсов ввода решений на Java. К тому же имеющийся в распоряжении разработчиков системы ряд педагогических программ, написанных на Java (WiseTasks [4], Verifier [5]), без особых трудностей может быть трансформирован в модули системы DCES.

Другая особенность соревнований, которую легко поддерживать в системе DCES – это разнообразие способов отображения условий. Возможностей некоторых систем не всегда хватает, чтобы задать вопрос таким образом, как этого хочется. Даже в популярном сервисе Google Документы при создании опроса можно задавать только вопросы, являющиеся короткими строками текста. Вставить в вопрос хотя бы картинку на момент написания статьи было невозможно. Тем не менее, условием математической задачи может быть формула, геометрический чертеж. А в задачах по программированию может появиться желание вставить текст программы, причем в этом случае

желательна автоматическая подсветка синтаксиса. Поэтому многообразие способов отображения условий является необходимым для поддержки различных типов соревнований.

Следующая особенность соревнований, которой позволяет управлять DCES – это правила отсылки решений. Настройки включают в себя время соревнования, в течение которого можно посылать решения, выбор ограничения на количество попыток, в течение которых можно посылать решение одной и той же задачи. Существующие системы достаточно хорошо справляются с настройками этих правил, поэтому нет необходимости подробно останавливаться на этом вопросе.

Из-за разнообразия способов ввода решений необходимо уметь поддерживать и достаточное разнообразие способов проверки решений. Например, если ответом в задаче является формула, то появляется задача сравнения присылаемых участниками формул с правильной. Это не может быть сделано только посимвольным сравнением формул. Если не усложнять условие задачи дополнительными требованиями по формату ввода ответа, для сравнения формул в общем случае необходима система компьютерной алгебры. DCES позволяет расширять возможности проверки, добавляя дополнительные модули. Так же, как это было в случае ввода решений, модули могут быть созданы сторонними разработчиками, но для большинства ситуаций базовая поставка системы уже содержит все наиболее часто используемые модули.

Режим доступа к результатам соревнования – это настройки, определяющие каким участникам и в какие моменты доступны те или иные результаты участников. В некоторых случаях участникам разрешается смотреть результаты других участников, в других – не разрешается смотреть даже свои [6, 7]. Режим доступа может изменяться со временем, например, после окончания соревнований таблица результатов обычно доступна всем.

Основные технические трудности, рассмотренные далее, будут связаны с последней особенностью соревнований – с процессом подведения итогов. Соревнования различаются устройством таблицы результатов, алгоритмом объединения результатов по отдельным задачам в общий результат и алгоритмом упорядочения участников в соответствии с их результатами. Форматы соревнований могут различаться настолько сильно, что учесть заранее все возможные алгоритмы представляется невозможным. Разнообразие алгоритмов поддерживается в системе возможностью подключать дополнительные модули с новыми алгоритмами. Тем не менее, при проектировании системы было решено избавить пользователя от необходимости использовать модули для большинства стандартных ситуаций и заставить систему самостоятельно понимать, какие алгоритмы должны быть применены. Например, если общий результат соревнования представляется в таблице одним столбцом с заголовком «баллы», очевидно, что участников нужно отсортировать по убыванию числа, и отдельно настраивать это поведение не нужно. Если же, например, в аналогичной ситуации в заголовке столбца написано «штрафные баллы», сортировать участников необходимо по возрастанию.

#### **ПРИМЕРЫ НАИБОЛЕЕ РАСПРОСТРАНЕННЫХ ТИПОВ СОРЕВНОВАНИЙ**

Обсудим несколько наиболее типичных примеров различных соревнований, на которые мы в дальнейшем будем ссылаться, обсуждая процесс подведения итогов. Все перечисленные соревнования могут быть проведены с помощью DCES, хотя это не означает, что авторы системы планировали использовать ее именно и исключительно для этих конкретных типов соревнований. В первую очередь, при описании форматов соревнований нас будет интересовать только та их часть, которая связана с подведением итогов.

#### **ОБЫЧНОЕ ТЕСТИРОВАНИЕ**

В обычном тестировании участники решают несколько задач, и для каждой задачи проверяющая система определяет, решена она или нет. Далее, каждой решенной задаче приписывается 1 балл (или больше), нерешенной – 0, все баллы складываются и полученная сумма баллов дает результат тестирования. Обычно при тестированиях участники не знают результаты друг друга, но, если необходимо создать общедоступную таблицу результатов, то она будет отсортирована по убыванию количества набранных баллов.

#### **ОБЫЧНАЯ КОНТРОЛЬНАЯ**

Обычная контрольная обобщает обычное тестирование. Здесь приписанные баллы у каждой задачи могут иметь не только значения 0 или 1, а выбираются из более широкого набора значений. Например, каждая задача может оцениваться баллами от 0 до 7, как это принято на Всероссийских и Международных математических олимпиадах. Помимо этого, в результатах контрольной, кроме суммарного балла, может содержаться и информация об интегральной оценке участника: отлично, хорошо, удовлетворительно, неудовлетворительно. Частным случаем обычной контрольной является тестирование, в котором за неправильный ответ ученик получает отрицательное число баллов, а за ненаписанный ответ – ноль.

#### **ПИСЬМЕННАЯ МАТЕМАТИЧЕСКАЯ ОЛИМПИАДА**

Так как сейчас мы обсуждаем только особенности олимпиад, связанные с подведением итогов, письменная математическая олимпиада для нас будет практически совпадать с обычной контрольной. Единственное отличие – во всех результатах проверки задач кроме количества баллов сохраняется запись о персональных данных или идентификационном коде члена жюри, который проверял это решение. Эта запись формально является частью результата проверки решения, хотя она и не влияет на окончательный результат.

## КОНКУРС КИО

В конкурсе КИО участники решают оптимизационную задачу. Решением, которое посылается на проверку, является сконструированный объект, а результатом проверки решения – значение оптимизационной функции. Например, участники могут строить ломаную, удовлетворяющую определенным ограничениям, и имеющую наибольшую длину. При подведении окончательных итогов каждый участник получает за задачу количество баллов, равное количеству других участников соревнования, которые решили задачу хуже него (в приведенном примере – построили более короткую ломаную). Далее баллы по задачам суммируются и участники сортируются по убыванию.

## ОЛИМПИАДА ПО ПРОГРАММИРОВАНИЮ

Олимпиады по программированию ACM ICPC имеют сложный формат проведения. Охарактеризуем кратко процесс подведения итогов. Результатом проверки каждой задачи является вердикт, решена она или не решена, время момента отсылки решения, номер отсылки, описание ошибки (неправильный ответ, ошибка компиляции) и номер теста, на котором произошла ошибка. При подведении итогов участник получает по баллу за каждую решенную задачу, и получает штрафные баллы, которые вычисляются для каждой решенной задачи следующим образом: время в минутах, прошедшее от начала олимпиады до отправки верного решения, плюс по двадцать штрафных баллов за каждую неверную попытку сдачи решения.

## ТУРНИР ПРОГРАММИСТОВ

В турнире программисты пишут программы, которые будут оцениваться в процессе «борьбы» между собой. При этом возможны ситуации, что программы побеждают друг друга по циклу, это означает, что их трудно упорядочить в линейный рейтинг и необходимо пользоваться другими системами проведения соревно-

ваний, например, круговыми турнирами. Подобная олимпиада (хотя она и может быть проведена средствами DCES) на самом деле является нехарактерной для применения такой системы. Для ее организации требуются неестественные решения, которые не будут работать в определенных ситуациях. Поэтому такой тип олимпиад мы обсуждаем, чтобы получить представление о границах возможностей системы. Пока что ограничение системы DCES можно сформулировать следующим образом: она позволяет проводить только соревнования, в которых решение каждой задачи может быть оценено независимо для каждого участника.

## ПРОЦЕСС ОПРЕДЕЛЕНИЯ РЕЗУЛЬТАТОВ

При определении результатов соревнования, решения участников проходят несколько этапов:

1. Оценка решения.
2. Преобразование результатов.
3. Определение окончательных результатов.
4. Сортировка.

На первом этапе у участников оцениваются решения задач. На втором этапе результаты преобразуются для записи в таблицу результатов, на третьем этапе заполняются окончательные результаты участников, на последнем этапе участники сортируются.

Для лучшего понимания разделения между первыми двумя этапами, обсудим его сейчас. Результат проверки задачи может содержать много информации, которая не должна быть доступна соперникам. Например, в соревновании по программированию при проверке решения определяется тип произошедшей ошибки – неверный ответ, превышен лимит времени и т.п. Участник знает эту информацию, но она недоступна другим участникам. Они знают только, решена ли задача, и эта информация доступна им через таблицу результатов. Таблица результатов содержит проекцию информации из результатов проверки.

## ОЦЕНКА РЕШЕНИЯ

После того как участник посылает решение в систему, в истории посылок появляется запись. Она содержит информацию об участнике, решенной задаче, времени отправки, посланном решении и, самое важное, о результате проверки. Сразу после отправки результат не заполняется, он появится только после того, как модуль проверки завершит работу. В некоторых случаях проверка происходит без задержки, например, для задач, в которых посланный ответ достаточно сравнить с правильным. В других случаях проверка может завершиться даже после окончания олимпиады, например, если все решения проверяются членами жюри вручную.

Результатом работы модуля проверки является несколько пар «ключ–значение». Например, после проверки задач обычных тестирований появляется одна пара с ключом «принято» и значением 0 или 1. В задачах контрольной создается пара «баллы» со значением, соответственно, количество баллов. В случае письменной олимпиады по математике, кроме пары «баллы», появляется пара «член жюри», в которой в качестве значения хранится информация о члене жюри, проверившем решение. В олимпиаде по программированию создается больше пар с результатами. Это пара «принято», пара «ошибка», содержащая информацию о произошедшей ошибке (ошибка компиляции, превышено время или память и т. п.), пара «тест», содержащая номер теста, на котором произошла ошибка.

Одна задача внутри системы DCES может переиспользоваться в разных соревнованиях. Но набор пар, который выдается при проверке ее решения, является характеристикой самой задачи и не может изменяться со временем.

Обсудим вопросы разграничения доступа к результатам соревнований. Система DCES позволяет настраивать доступ так, что в каждый момент времени любой участник имеет доступ либо только к своим результатам (режим 1), либо ко всем

результатам других участников (режим 2), либо не имеет доступа к результатам вообще (режим 0). Режим 0, например, имеет смысл в случае тестирований, когда до окончания тестирования участник не знает свои результаты. Эти настройки относятся, в первую очередь, к таблице результатов, которую мы обсудим в следующем разделе. Но эти же настройки влияют и на доступность результатов, создаваемых модулями проверки решений.

Результаты, выданные модулем проверки решений, доступны участнику, только если текущий режим для него в данный момент 1 или 2. Другими словами, результаты проверки он может смотреть, только если ему в принципе разрешен доступ к своим результатам. Участнику никогда не доступны результаты проверки решений других участников, например, в олимпиаде по программированию не следует сообщать соперникам, на каких тестах «сломались» решения участника. Соперники должны знать только, решена задача или нет. Как уже было сказано, они узнают это из таблицы результатов.

Считается, что результаты проверки решения должны быть либо недоступны участнику (в режиме 0), либо доступны полностью. Другими словами, жюри не должно хранить в результатах проверки то, что оно не хочет ему сообщать. Во время олимпиады и после нее участник имеет доступ к своей истории посылок и может смотреть все данные из результатов проверки.

Это ограничение системы DCES можно обойти, если зашифровать данные проверки. Например, если жюри не желает сообщать конкретного члена жюри, проверявшего задачи, она может хранить не его имя, а его номер.

## ПРЕОБРАЗОВАНИЕ РЕЗУЛЬТАТОВ

Результаты проверки решения участника доступны членам жюри и самому участнику (в режиме 1 или 2). Чтобы делиться результатами с соперниками, необходимо заполнить таблицу результатов.

Таблица результатов для каждой задачи и каждого участника содержит набор пар «ключ–значение». В простейшем случае это те же самые пары, что были созданы при проверке. Например, в обычных тестированиях результат проверки содержит только одну пару с ключом «принято», и эта же пара должна появиться в таблице результатов.

Участник может посылать решение несколько раз, но в таблице результатов должна появиться только одна запись. Администратору соревнования необходимо сделать настройки в соответствии с тем, какой именно результат проверки попадет в таблицу. Существует всего две настройки – последняя посылка или лучшая. Определением, какая посылка лучше, занимается модуль проверки решений.

В общем случае настроить преобразование пар из результатов проверки в пары для таблицы результатов можно произвольным сколь угодно сложным образом. Но универсальной системой, в которой требуется настраивать все, было бы очень трудно пользоваться. Поэтому DCES в определенных случаях способна сама догадаться, как преобразовать набор пар.

Например, если в результате проверки задачи олимпиады была создана пара «баллы» и пара «член жюри», DCES способна автоматически распознать, что в таблицу результатов необходимо перенести только пару «баллы» с неизменным значением.

Если дополнительных настроек нет, для преобразования используется следующий алгоритм: в результате проверки анализируются ключи. Некоторые ключи известны системе и помечены как «переносимые». Например, переносимыми являются ключи **accepted** (принято), **scores** («баллы») **penalty** («штрафные баллы»). Если найдены переносимые ключи, то все пары с этими ключами и только они переносятся в таблицу результатов. Если переносимых ключей не найдено, в таблицу результатов переносятся все пары.

Это не единственное место, где ключи имеют смыслы и влияют на поведение си-

стемы. Существует ряд ключей, рекомендованных для использования в модулях проверки, они имеют определенный смысл для разных этапов подведения результатов соревнований. Например, если при проверке решения определяется только решена задача или нет, рекомендуется использовать ключ **accepted**, если при проверке считаются баллы – ключ **scores**.

В случае, если действия системы по умолчанию не устраивают администратора соревнования, он может настроить преобразование результатов для каждой задачи отдельно. Для этого задаются набор инструкций, имеющих вид:

```
rt.scores = chk.accepted * 5
rt.time = elapsedTime
```

Приведенные в примере инструкции означают, что таблица результатов имеет две пары: **scores** и **time** (время). Префикс **rt** (от results table) задает ключ из таблицы результатов. Префикс **chk** (от check) задает ключ из результатов проверки. Значение ключа **scores** это значение ключа **accepted**, умноженное на 5. Вторая инструкция указывает, что в таблице результатов необходимо создать пару **time**. Это время, прошедшее с начала соревнования. Вычисление времени происходит с помощью встроенной функции **elapsedTime**. Первое правило, приведенное в примере, можно интерпретировать так: составитель соревнования желает давать за решение задачи 5 баллов. Возможно, в соревновании будут участвовать и другие задачи с другим количеством баллов.

Тот факт, что в преобразовании результатов создается пара «время», означает, что жюри желает принимать его во внимание при подведении окончательных итогов. Если некоторые данные необходимо учитывать при подведении итогов, их требуется перенести в таблицу результатов. Это требование кажется естественным, иначе окажется, что итоги олимпиады не соответствуют таблице результатов.

Описанные выше возможности настройки преобразования результатов можно расширить с помощью дополнительных

модулей. Описанный алгоритм и интерпретация инструкций по преобразованию являются функциональностью основного модуля преобразования результатов. Другие модули тоже могут быть реализованы, но необходимость в этом еще ни разу не возникала.

#### ОПРЕДЕЛЕНИЕ ОКОНЧАТЕЛЬНЫХ РЕЗУЛЬТАТОВ И СОРТИРОВКА

Первые два этапа определения результатов могут происходить только на стороне сервера соревнований. Именно сервер проверяет решения, и он же вычисляет значения для таблицы результатов. Далее, на основе таблицы вычисляются окончательные результаты, и производится сортировка участников. Но таблица результатов содержит общедоступные данные, к тому же, правила вычисления окончательных результатов тоже должны быть общедоступны. Следовательно, вычислить окончательный результат и отсортировать участников можно не только на стороне сервера, но и на компьютере участников. Именно такой подход применен в текущей реализации системы DCES, в частности для разгрузки сервера и упрощения его кода. Обсуждение некоторых других вопросов, связанных с безопасностью систем с автоматизированной проверкой решений, можно найти в [9].

Как и на предыдущих этапах подведения итогов, процесс определения окончательных результатов для каждого участника может быть устроен сколь угодно сложно. Различные алгоритмы реализуются с помощью разных подключаемых модулей. Но в стандартных случаях администратору соревнования не нужно думать о настройках, система все сделает сама.

Окончательные результаты участника – это набор пар «ключ–значение». Например, результаты контрольной состоят из двух пар `scores` и `mark` (оценка). Баллы являются суммой баллов по отдельным задачам, оценка выставляется в зависимости от количества баллов.

В простых случаях достаточно полагаться на самостоятельное поведение системы. Например, если результаты по каждой задаче состоят только из ключей `scores`, то окончательный результат тоже будет состоять только из ключа `scores`, значение которого будет суммой баллов по отдельным задачам. Отсортированы результаты будут по убыванию количества баллов.

В общем случае при отсутствии настроек используется следующий алгоритм вычисления пар ключей и значений для окончательного результата: в каждой задаче анализируется набор ключей. Среди них ищутся те, которые известны системе как «суммируемые». Это ключи `scores`, `accepted`, `penalty`, `time` и другие. Если некоторый суммируемый ключ найден хотя бы в одной задаче, пара с этим ключом будет использоваться в окончательном результате. Значение, присвоенное каждому ключу, является суммой значений ключей по отдельным задачам. В случае отсутствия в определенной задаче ключа, его значение принимается равным нулю.

Сортировка результатов, применяемая по умолчанию, работает за счет того, что для всех суммируемых ключей система знает их стандартное направление сортировки и приоритет. Например, заголовки `scores`, `accepted` сортируются по убыванию, `penalty` и `time` – по возрастанию. При этом, если результат содержит несколько ключей, в первую очередь сортировка происходит по заголовку `accepted`, при прочих равных по заголовку `scores` и т.д.

Эту функциональность реализует модуль вычисления результатов, подключенный по умолчанию. Существует возможность воспользоваться другим модулем, настроив его набором инструкций, похожих на инструкции по преобразованию результатов. Они описывают множество ключей в окончательном результате и способ сортировки участников. Приведем пример инструкций, который можно использовать в контрольной, если помимо количества баллов требуется сообщить участнику полученную им оценку.



```
res.scores = sum(rt.scores)
res.mark = res.scores > 50 ?
    "отлично" : res.scores > 40 ?
    "хорошо" : res.scores > 30 ?
    "удовлетворительно" :
    "неудовлетворительно"
sort = "scores"
```

Префикс `rt`, как и раньше, означает ссылку на данные из таблицы результатов. `rt.scores` означает список, составленный из значений по отдельным задачам, функция `sum()` суммирует список. Кроме `sum()` существуют несколько других встроенных функций, например, функция `exceeds()`, она считает количество результатов других участников, которые меньше результата этого участника. (Можно использовать при вычислении результатов в конкурсе КИО).

Префикс `res` означает ссылку на окончательные результаты участника. При заполнении `mark` (оценки) вычисляется выражение, содержащее оператор `?:` он знаком программистам C, Java, JavaScript и других Си-подобных языков.

Последняя строка указывает заголовок для сортировки. Указать можно несколько заголовков или даже задать собственную функцию сравнения.

Может показаться, что настройка программы с помощью написания программы недоступна большинству пользователей, особенно тем, кто не является программистами. И что удобный пользовательский интерфейс был бы лучшим решением. Но опыт показывает, что пользователи способны написать простую и короткую программу, если рядом с полем ввода приведены примеры коротких программ с пояснениями. Приведенный выше пример является далеко не самым тривиальным, потому что при вычислении оценки используется сложное выражение. Но такие выражения встречаются нечасто, обычно необходимо лишь указать набор столбцов и правила сортировки. Описанный язык является примером предметно-ориентированного языка, возможности его использования ограничены описанием правил преобразования результатов внут-

ри системы. Подробнее о предметно-ориентированных языках можно найти в [8].

После того как мы обсудили все стадии вычисления результатов, посмотрим, подходят ли они для того, чтобы реализовать соревнование «турнир программистов». На первом этапе, то есть при проверке посланного решения, ничего сказать о качестве этого решения нельзя. Его необходимо сравнить с решениями других участников, которые на этом этапе недоступны. Можно лишь проверить, например, компилируются ли присланные решения.

Чтобы сравнить решения участников друг с другом, их необходимо напрямую скопировать в таблицу результатов. Такое поведение может показаться странным, но если подумать, никаких ограничений на содержание таблицы наложено не было. Итак, один участник в таблице результатов имеет запись об одной задаче (будем считать, что соревнование состоит только из одной задачи), и в этой записи хранится его программа.

На третьем этапе при вычислении окончательных результатов и происходит сам турнир. Необходимо создать модуль подведения итогов, который запустит программы участников друг против друга. Такой модуль можно описать на приведенном выше языке, но можно и создать отдельно.

Несмотря на то, что существует теоретическая возможность проводить турниры программистов, система DCES для них недостаточно приспособлена. В первую очередь, необходимо самостоятельно реализовать модуль подведения итогов, уступающий сам турнир. Во-вторых, тот факт, что программы участников хранятся в общедоступной таблице результатов, означает, что участники могут читать программы друг друга. Проблем можно избежать, компилируя программы перед записью в таблицу результатов или изменив режим доступа к результатам на 1. Но при этом режиме полная таблица результатов доступна только членам жюри, поэтому им необходимо публиковать результаты уча-

стников каким-то другим способом, не связанным с системой.

### ОФОРМЛЕНИЕ РЕЗУЛЬТАТОВ

Все обсуждение выше было связано с представлением результатов внутри системы и стадиями их преобразований. Но пользователи системы, то есть администраторы соревнований и участники, должны видеть результаты в удобной для восприятия форме. Как и все другие аспекты поведения системы, отображение результатов может быть настроено произвольным сколько угодно сложным образом. При этом для частых ситуаций система должна работать и при отсутствии любых настроек. Базовый способ отображения таблицы результатов, если система не смогла применить ничего другого, следующий: таблица результатов отображает участников по строкам, задачи и окончательный результат – по столбцам, все столбцы состоят из подстолбцов, соответствующим ключам в результатах проверки. Но несколько подстолбцов занимают много места, и такая таблица будет трудна для восприятия. Поэтому для некоторых наборов из подстолбцов в системе предусмотрены специальные способы отображения. Например, если задача содержит только подстолбец **accepted**, то в соответствующей клетке таблицы появляется не число 1 или 0, а, соответственно, символ плюс или ничего. Если задача состоит из подстолбцов **accepted** и **penalty**, то в соответствующей клетке таблицы появляются плюс, и *под ним* пишется количество штрафных баллов. Наборы подстолбцов, распознаваемые системой, можно добавлять и изменять с помощью дополнительных модулей. Для создания модуля достаточно описать собственную реализацию интерфейса `TableCellRenderer`, которую система будет использовать для рисования ячеек таблицы.

Если набор подстолбцов не распознан, они будут нарисованы в таблице как есть с соответствующими заголовками. Обычно заголовки пишутся не напрямую, а переводятся на русский язык. Например, система пишет «баллы» вместо «scores» и «время» вместо «time».

Кроме таблицы результатов, участники способны просматривать свою историю посылок решений по каждой задаче. Отображение истории происходит, так же как и в случае таблицы результатов, но распознавания набора столбцов не происходит. Таким образом, для каждой задачи участник видит одну таблицу, по строкам расположены посылки, по столбцам – ключи, помечающие значения с информацией о посылке и результатах ее проверки.

### ЗАКЛЮЧЕНИЕ

Мы рассмотрели систему проведения удаленных соревнований DCES и, конкретно, ту ее часть, которая связана с процессом определения результатов соревнований. Процесс состоит из четырех этапов. На первом этапе проверяются решения, присылаемые участниками. На втором, в зависимости от настроек, в таблицу результатов переносится либо лучший результат, либо результат последней проверки, при этом при переносе в таблицу результаты преобразуются по заданным правилам. На третьем этапе формируются окончательные результаты каждого участника, и на четвертом участники сортируются в соответствии с полученными результатами.

Процесс получения окончательных результатов может быть сколь угодно сложным, на каждом этапе на него могут влиять дополнительные подключаемые модули. При этом система способна работать без каких-либо дополнительных настроек, пытаясь определить необходимое поведение по ключам обрабатываемых данных.

## Литература

1. *Посов И.А., Рукишин С.Е.* Генерируемые задачи в системе для организации удаленной работы с задачами // Научно–технический вестник СПбГУ ИТМО, 2010. № 5(69). С. 130.
2. *Посов И.А., Рукишин С.Е.* Архитектура системы проведения удаленных соревнований и организации работы с математическими задачами // Научно–технические ведомости СПбГПУ, 2010. № 3(101). С. 49–53.
3. *Рукишин С.Е.* Классификация типов научных соревнований с автоматической обработкой решений // Научно–технический вестник СПбГУ ИТМО, 2010. № 3(67). С. 121–125.
4. *Богданов М.С.* Автоматизация проверки решения задачи по формальному описанию ее условия // Компьютерные инструменты в образовании, 2006. № 4. С. 51–57.
5. *Манцеров Д.И.* Система верификации для параметрических классов задач по математике // Научно-технические ведомости СПбГПУ, 2008. № 5. Информатика. Телекоммуникации. Управление. Федеральное агентство по образованию. Санкт-Петербургский государственный политехнический университет, 2008. С. 183–1896.
6. *Рукишин С.Е.* Сравнительные достоинства и недостатки дистанционных и традиционных олимпиад и их влияние на архитектуру автоматизированных систем поддержки дистанционных научных соревнований // Международный журнал Образовательные технологии и общество, 2010. Т. 13. № 3. С. 347–359.
7. *Рукишин С.Е.* Основы конструирования систем проведения дистанционных научных соревнований // Программные продукты и системы», 2010. № 3(91). С. 69–72.
8. *Фаулер М.* Языковой инструментарий: новая жизнь языков предметной области // <http://www.maxkir.com/sd/languageWorkbenches.html>. [Электронный ресурс] / MAXKIR.com. Самое интересное о разработке программного обеспечения – Электрон. дан. – 27.12.2005 – Режим доступа : <http://www.maxkir.com/sd/languageWorkbenches.html> свободный, – Загл. с экрана. – Яз. рус./ (последнее обращение: 06.10.2010).
9. *Дементьева О.М., Кавецкий С.Ф., Кручинин В.В.* Проблемы и пути решения защиты учебных программ // Современные средства и системы автоматизации. Томск: изд-во Томск. гос. ун-та систем упр. и радиоэлектроники, 2004. С. 183–185.

## Abstract

The paper discusses the evaluation of problems solutions in the DCES System (Distance Competitive Educational System). The system is used to handle distance competitions and tests with the automated solutions checking and automated obtaining of results. Evaluation of participant solutions occurs in four steps: an objective assessment, information transfer to the results table, the overall results summarization, sorting of participants. Each step with all the possibilities to configure it is described.

**Keywords:** Education, Distance Competitive Educational System, automated testing, results summarization.

*Рукишин Сергей Евгеньевич,  
доцент кафедры математического  
анализа РГПУ им. Герцена,  
vliuser@gmail.com*



Наши авторы, 2010.  
Our authors, 2010.