С. Е. Рукшин

ПРИНЦИПЫ И ТЕХНОЛОГИИ ИНТЕГРАЦИИ МАТЕМАТИЧЕСКИХ ЗАДАЧ В СИСТЕМУ АВТОМАТИЗИРОВАННОЙ ПОДДЕРЖКИ НАУЧНЫХ СОРЕВНОВАНИЙ

Развитие дистанционных форм обучения делает актуальной задачу развития автоматизированных систем поддержки дистанционных математических соревнований. Для того чтобы сохранить накопленные традиции в области проведения научных соревнований, необходимо учесть два фактора: многообразие форм задач и многообразие регламентов различных конкурсов. В статье представлены принципы, технологии и конкретные примеры интеграции математических задач различных типов в системы с генерацией вариантов задач и автоматической проверкой решений и ответов. Также в статье рассматриваются структура и функционирование компьютерной системы поддержки математических соревнований, предназначенной как для проведения дистанционных соревнований по математике и другим предметам естественно-математического цикла, так и для поддержки дистанционной подготовки по математике посредством решения различных типов математических задач.

Ключевые слова: дистанционное обучение, математические соревнования, автоматизированные системы в обучении.

S. Rukshin

PRINCIPLES AND TECHNOLOGY OF THE INTEGRATION OF MATHEMATICAL PROBLEMS IN THE SYSTEM OF AUTOMATIC SUPPORT OF SCIENTIFIC COMPETITIONS

The development of distance education puts forward the problem of constructing automatic systems for the support of remote mathematical competitions. There are two essential factors for the construction which must be taken into account: the diversity of challenges and the diversity of regulations of the competition. The principles, the technology and specific examples of the integration of mathematical problems of various types into the system with automatic verification of problem solutions are presented. The structure and functioning of the computer support system designed both for distance competitions in mathematics and for distance education through problem solving are also discussed.

Keywords: distance education, mathematical competitions, automatic systems.

Введение

В работе [9], обобщающей мировой опыт развития стимулирующих занятий математикой [13], в качестве одного из важнейших направлений развития информационно-коммуникационных технологий для поддержки увлеченности школьников математикой рассматривается конструирование так называемых ЧРП — числовых рабочих пространств (Numerical Working Spaces). В основе концепции ЧРП лежит понятие общего информационного пространства, которое сформулировано в работе [11] и применительно к использованию в изучении и преподавании математики развито в работах [3, 6]. Другой основой создания ЧРП является теория конструирования информационной среды обучения [1].

В качестве характерных признаков ЧРП можно отметить:

– наличие программного обеспечения для проведения экспериментов и моделирования;

– наличие баз данных, например банков задач;

– числовые рабочие пространства — место для обсуждения решений математических задач [13].

Таким образом, эти пространства можно приспособить ко всем вариантам занятий математикой, независимо от их формы, дать любому человеку возможность познакомиться с новыми математическими идеями, углубить математические знания и провести экспертизу своих знаний. Числовые рабочие пространства это хороший инструмент для подготовки учебных групп или отдельных школьников к математическим соревнованиям [13].

В то же время пока в использовании общих информационных пространств для поддержки обучения математике имеются нерешенные проблемы. К их числу можно отнести:

– неспособность большинства школьных задач «из учебника» математики вызвать плодотворное обсуждение (содержательная проблема);

– ограниченные возможности используемых инструментов для передачи знания (технологическая проблема) [14].

Для решения этих проблем предлагаются инновационные подходы, допускающие использование серьезных математических задач, которые вовлекали бы обучаемых в моделирование, выявляющее математическую сущность проблемы; способствовали бы развитию новых инструментов, позволяющих по-новому представить математические проблемы и организовать конструктивный диалог с учеником, поддерживаемый компьютерными средствами [14].

Предлагаемая в статье система демонстрирует возможные пути решения указанных проблем.

Типология задач

Обозначенную выше проблему будем решать посредством расширения способов работы с математической задачей в системах компьютерной поддержки решения математических задач. Для этого рассмотрим метод, предложенный в работах [5, 12]: использование различных представлений информации для расширения спектра заданий по математике. Задачи в этой интерпретации рассматриваются как перевод с одного языка представления информации на другой.

В качестве достаточно общего вида таких задач можно выделить задачи на описание множества одним из двух способов по описанию другим: описание предикатами и конструктивное описание.

Обратим внимание, что к этому виду относятся практически все «стандартные» задачи, сформулированные в форме «дано… (предикатное описание) — найти… (конструктивное представление)». В то же время обратная задача также представляет большой интерес для развития новых технологий обучения математике. Это направление разрабатывается в работах С. Г. Иванова [4].

Рассмотрим более подробно технологические аспекты использования таких задач (обозначим их как **вид 1**)в системе дистанционной поддержки математических соревнований на примере задач по комбинаторике, обобщив и расширив результаты работы [2].

Выделим в этом виде задач следующие подвиды и типы.

Подвид 1-1. Комбинаторные задачи на подсчет числа вариантов, в которых по словесному описанию множества нужно написать формулу с переменной либо с числовыми значениями.

Тип 1. Комбинаторные задачи с видом ответа — формула (формулу можно рассматривать как частный случай алгоритма: линейный алгоритм с использованием типовых комбинаторных функций).

Пример задачи типа 1. Трамвайный билет называется счастливым по-питерски, если сумма первых трех его цифр равна сумме трех последних цифр. Трамвайный билет называют счастливым по-московски, если сумма его цифр, стоящих на четных местах, равна сумме цифр, стоящих на нечетных местах. Сколько существует билетов, счастливых и по-питерски, и по-московски, включая билет 000000?

Замечания по реализации. Решение задачи сводится к описанию алгоритма вычислений, который, как во многих комбинаторных задачах, определяется формулой вычисления, содержащей натуральные числа, арифметические операции, функции «факториал» и «число сочетаний» и, возможно, функцию ввода знакопеременной суммы с общим членом для использования формулы включений/исключений. Формула вводится через клавиатуру, содержащую базовый набор комбинаторных функций.

В работах И. А. Посова и А. В. Степанова [8, 10] показано, какой методический потенциал таит в себе параметризация математических задач, которая позволяет обеспечить формирование базовых умений в рамках как высшего [10], так и среднего [8] образования. В то же время создание параметрических генераторов является отдельной методической и технологической задачей. Для достижения целей нашей работы следует отметить важность параметрических генераторов для проведения научных олимпиад и экзаменационных мероприятий: это позволяет уменьшить возможности для несамостоятельного выполнения заданий, а также обеспечить тренировку в решении математических задач при подготовке к олимпиадам или сдаче экзаменов.

Рассмотрим на приведенном выше примере возможную параметризацию этой задачи:

Параметризация. Ввести параметр $2n$ — длина набора. Возможен еще один параметр — p, определяющий основание системы счисления, в которой задается билет.

Наличие параметра в задаче дает еще одну важную возможность для организации дистанционных соревнований. Особенно это значимо для соревнований «нижнего» уровня, т. е. проведения соревнований в школе. В этих соревнованиях требуется стимулировать к решению математических задач большую часть учеников. Один из путей реализации этого требования — организация диалога с учеником. Такой способ представления математических задач разработан в работах С. Н. Позднякова, С. Г. Иванова, Д. И. Манцерова [1, 4, 5, 7, 12]. В них представлены различные аспекты (теоретические, методические, технические) использования механизма верификации решений задач, содержащих параметры, на множестве генерируемых примеров и организации корректирующего диалога в процессе решения задачи.

Для нашего примера верификация реализуется следующим образом:

Верификация решения. При параметризации решение проверяется для различных значений параметров и принимается, если все тесты прошли успешно.

По составленной схеме проанализируем другие типы задач.

Тип 2. Комбинаторные задачи с промежуточным этапом решения, предполагающим описание решения предикатом, и конечным этапом, сводящимся к типу 1.

Пример задачи типа 2. Дан куб $12 \times 12 \times 12$, который разрезан плоскостями, параллельными граням куба, на единичные кубики. На сколько частей разделится куб, если в нем провести сечение в виде правильного шестиугольника?

Замечания по реализации. Решение задачи сводится к решению в неотрицательных целых числах, не превосходящих 11, уравнений $a + b + c = 16$ и $a + b + c = 17$. Ответ можно вводить в виде числа либо в виде предиката. В обоих случаях задача сохранит содержательность. Возможно введение двух этапов решения: описание предикатом (идея решения — изоморфизм) и описание алгоритмом вычисления (числом, которое может вводиться через клавиатуру, содержащую базовый набор комбинаторных функций — факториал, число сочетаний и арифметические операции).

Параметризация. Ввести параметр n — длина ребра. Возможен еще один параметр — m, определяющий вид сечения.

Замечания по верификации ответа. При параметризации решение проверяется для различных значений параметра и принимается, если все тесты прошли успешно.

Тип 3. Комбинаторные задачи с промежуточным этапом решения, предполагающим описание решения рекуррентной формулой, и конечным этапом, сводящимся к типу 1.

Пример задачи типа 3. Имеется k красок. Сколькими способами можно раскрасить стороны данного правильного n-угольника так, чтобы соседние стороны были окрашены в разные цвета (многоугольник поворачивать нельзя).

Замечания по реализации. Решение задачи связано с использованием метода математической индукции. Поэтому частью решения является написание рекуррентных формул. В данном случае это формулы: $X[1] = 0$; $X[2] = k(k - 1)$; $X[n] = (k - 2)X[n - 1] + (k - 1)X[n - 2]$. Далее по индукции доказывается явная формула $X[n] = (k - 1)$^$n + (k - 1)(-1)$^n.

Задачу можно разбить на этапы. На первом этапе предлагается виртуальная клавиатура, не содержащая степени, но имеющая клавиши для ввода рекуррентных соотношений. На втором этапе используется другая клавиатура: без средств для ввода рекуррентных соотношений, но со средствами ввода любых элементарных функций, которые должны быть реализованы для натуральных чисел произвольной длины.

Параметризация. Естественная параметризация: параметр k — число красок, параметр n — число сторон многоугольника.

Замечания по верификации ответа. На первом этапе решение проверяется на последовательностях для различных значений параметра k, на втором этапе сравниваются две функции для различных значений параметров k и n. Решение принимается, если все тесты прошли успешно.

Тип 4. Комбинаторные задачи с видом ответа — формула или формулы, которые задают оценки числа вариантов сверху и снизу.

Пример задачи типа 4. В школе изучают $2n$ предметов. Все ученики учатся на «четыре» и «пять» и никакие два не учатся одинаково. Ни про каких двух учеников нельзя сказать, что один учится лучше другого. Найдите как можно более хорошую оценку сверху для числа учеников в школе.

Замечания по реализации. Для ввода ответа нужна виртуальная клавиатура с описанными выше функциями и возможностями.

Параметризация. Параметризация неочевидна, необходимо объединять в один класс различные комбинаторные задачи.

Замечания по верификации ответа. Решение сравнивается с точным решением, подсчитывающимся для различных значений n по алгоритму, либо сравнивается с эталонной оценкой, предлагаемой в решении задачи. Решения разных участников могут сравниваться по степени приближения к точному решению.

Как уже отмечалось выше, большое значение имеет обращение задач. В данном случае это будут задачи, связанные с постановкой задач в общем виде на основе проведенных экспериментов, а также описание конструктивных решений, основанных на использовании предикатных конструкций.

В качестве примера рассмотрим следующий подвид комбинаторных задач.

Подвид 1-2. Комбинаторные задачи, в которых требуется описать множество вариантов, используя предикаты, задающие искомые наборы, либо алгоритмы, генерирующие эти наборы.

Тип 5. Комбинаторные задачи на перечисление вариантов, которые предполагают два этапа: первый — конкретный пример перечисления (для небольшого значения n); второй — формула, указывающая искомое сопоставление.

Пример задачи типа 5. В дом отдыха с четырехразовым питанием на 15 дней приехала группа отдыхающих из 60 человек. За обеденным столом 61 место. На одном месте постоянно сидит директор дома отдыха. Директор хочет сам познакомиться с каждым отдыхающим и познакомить их между собой. Для этого он предлагает отдыхающим садиться каждый раз по-новому, чтобы ни один из них не сидел дважды на одном и том же месте и чтобы у всех отдыхающих и у директора каждый раз был новый сосед справа. Как это сделать?

Замечания по реализации. Решение задачи записывается формулой: на $(k + 1)$ день отдыхающие садятся в порядке

k, k — 1, k — 1 + 2, … k — 1 + 2 — 3 + 4–…+(−1)^(i − 1), … (mod 60), где все отдыхающие занумерованы числами от 0 до 60, а номера мест нумеруются от директорского места по часовой стрелке. В решении задачи можно выделить два этапа: этап частных примеров с небольшим числом отдыхающих 2–3 человека. На этом этапе ответ вводится в графической форме — перемещением кружков по кругу. Второй этап — формализация ответа и переход к общему случаю. Для ввода ответа предлагается изоморфное описание ответа в виде правила сопоставления (номера отдыхающих — номера стульев). Для ввода формулы служит виртуальная клавиатура, включающая операцию суммирования конечной последовательности, операцию вычисления остатка, а также стандартные арифметические операции вместе с операцией возведения в степень. Операции выполняются над целыми числами произвольной длины.

Параметризация. Естественная параметризация: параметр n — число отдыхающих. Можно ввести «камуфлирующий» параметр s — число мест за столом ($s > n$). Число отдыхающих будет s — 1. Камуфлирующим будем называть параметр, не усложняющий решения задачи, но позволяющий сделать ее многовариантной.

Замечания по верификации ответа. Поскольку решений задачи может быть много, верификация осуществляется не сравнением с эталонным ответом, а проверкой выполнения условий задачи.

Подвид 1-3. Задачи, решение которых связано с описанием множества в терминах глобальных характеристик по описанию множества локальными характеристиками.

Тип 6. Описание языков предикатами по заданной грамматике.

Пример задачи типа 6. Назовем словом любую конечную последовательность букв А и Б. Есть две операции над словами: 1) в любом месте слова вставить А и в конце слова приписать Б; 2) в любом месте слова вставить АБ. Докажите, что слова, которые можно получить из слова АБ, используя только первую операцию, — это те же слова, которые можно получить из слова АБ, используя только вторую операцию.

Указание. Описать каждое из множеств слов, используя предоставленные инструментальные средства.

Замечания по реализации. Решением в обоих случаях является множество слов, в которых среди любых первых букв количество букв А не меньше, чем количество букв Б. Таким образом, для описания языка нужны средства подсчета числа букв. Например, функция $A(n, m)$, определяющая число букв А на промежутке от n-й до m-й букв. Кроме этого, нужны кванторы, чтобы записать предыдущее высказывание и «камуфлирующие» функции, расширяющие описательные возможности языка, например, $\mathrm{SAB}(n, m)$ — если на n-м месте стоит буква А, то на m-м месте стоит буква Б.

Параметризация. Параметризация осуществляется путем использования разных правил генерации слов и разных начальных слов.

Замечания по верификации ответа. Ответ проверяется по правилам грамматики, по которым генерируются слова, а затем подстановкой в предикат. Нужна обратная проверка, поэтому лучше сгенерировать все возможные слова заданной длины и проверить совпадение множеств.

Подвид 1-4. Конструирование геометрических объектов с заданными свойствами.

Тип 7. Геометрические задачи, решение которых определяется графом.

Пример задачи типа 7. Найдите на клетчатой бумаге фигуру, составленную из наименьшего числа клеток и обладающую следующим свойством: когда двое играют в крестики-нолики, то на клетках этой фигуры начинающий всегда может первым поставить три крестика подряд.

Замечания по реализации. Решением задачи является фигура из клеток. Внутреннее представление ответа — граф, который определяется вершинами, стоящими в выбранных клетках, и ребрами, соединяющими вершины в смежных (по сторонам клеток) клетках.

Параметризация. Задача параметризуется неочевидно, например изменением правил игры.

Замечания по верификации ответа. Можно организовать полный перебор вариантов игры с эвристическим выбором приоритета очередного хода.

Тип 8. Задачи на конструирование отношения на множестве с заданными свойствами (решение определяется некоторым графом).

Пример задачи типа 8. На площади собралось 50 гангстеров. Они одновременно стреляют друг в друга, причем каждый стреляет в ближайшего к нему или в одного из ближайших. Каково минимальное количество убитых? (Расположите на плоскости 50 точек, изображающих стреляющих, задайте расстояния между ними и укажите, кто в кого стреляет.)

Замечания по реализации. Интерфейс задачи — интерфейс для изображения ориентированного взвешенного графа.

Параметризация. Естественный параметр — число стреляющих гангстеров. Другие, менее очевидные возможности для варьирования — изменение правил, по которым идет стрельба.

Замечания по верификации ответа. Верификация осуществляется простой проверкой выполнения условий задачи.

Подвид 1-5. Конструирование многочленов с заданными свойствами.

Тип 9. Задачи на конструирование целых чисел с заданными свойствами в форме значений конкретных заданных многочленов.

Тип 10. Конструирование целых чисел с заданными свойствами в форме значений n многочленов, коэффициенты которых заданы функцией $a(n; k)$.

Пример задачи типа 9 и 10. $F1$, $F2$, $F3$ … Fn — многочлены с целыми коэффициентами. Докажите, что при некотором целом a [постройте a] все числа $F1(a)$, $F2(a)$, $F3(a)$ … $Fn(a)$ составные.

Замечания по реализациии задачи типа 9. Ввод ответа осуществляется через виртуальную клавиатуру, обеспечивающую возможность ввода функций, в данном случае — многочленов.

Замечания по реализациии задачи типа 10. Ввод ответа осуществляется через виртуальную клавиатуру, обеспечивающую возможность использования конечных сумм, зависящих от n и общего члена последовательности Fn, который рассматривается как функция.

Параметризация задачи типа 9. Параметризация осуществляется изменением числа многочленов n и коэффициентов многочленов.

Параметризация задачи типа 10. Параметризация осуществляется по общим формулам коэффициентов многочленов.

Замечания по верификации ответа. Для введенного числа *a* программа проверяет, являются ли все значения составными числами.

Вид 2. Задачи на составление алгоритмов решения математических задач.

Подвид 2-1. Задачи, решение которых описывается линейным алгоритмом.

Тип 11. Задачи на геометрические построения (например, задачи на построения циркулем и линейкой).

Пример задачи типа 11. Внутри правильного треугольника разместить треугольник с заданными углами так, чтобы он имел наибольшую площадь.

Замечания по реализации. Прототипом интерфейса может служить среда «The Geometer's Sketchpad» или 1С Математический конструктор. Интерфейс для решения задачи предполагает предоставление необходимых инструментов для выполнения построений (например, циркулем и линейкой).

Параметризация. Для генерации различных вариантов можно менять углы треугольника, который надо разместить в правильном треугольнике.

Замечания по верификации ответа. Можно использовать метод Монте-Карло. При этом его надо применить так, чтобы уменьшить число генерируемых треугольников. (Например, выбирать точку на отрезке, затем выбирать угол, под которым проводить луч, на котором лежит вторая вершина, второй луч определится автоматически и т. д.)

Подвид 2-2. Задачи, решение которых описывается ветвящимся алгоритмом. Комбинаторные задачи на перечисление, решение которых задается ветвящимся алгоритмом.

Тип 12. Задачи на «взвешивания».

Пример задачи типа 12. Есть *N* чеканщиков. Некоторые из них изготавливают только фальшивые монеты, а другие — только настоящие. Вес фальшивой монеты отличен от веса настоящей. Имеются весы с полным набором гирь и одна заведомо настоящая монета. У каждого чеканщика можно взять сколь угодно монет. Как с помощью минимального количества взвешиваний (трех) определить всех фальшивомонетчиков?

Замечания по реализации. Решение задачи записывается в виде нескольких шагов — взвешиваний, на каждом из которых задается (указанием функции от *n* — номера чеканщика) число монет от каждого чеканщика, участвующее во взвешивании, и ветвление, указывающее номер следующего шага в зависимости от результата взвешивания.

Параметризация. Естественная параметризация — от *N* — количества чеканщиков. Возможно введение различных ограничений (на число взвешиваемых монет, количество фальшивомонетчиков).

Замечания по верификации ответа. Алгоритм проверяется на множестве тестов — различных способах распределения чеканщиков (фальшивомонетчик или нет).

Подвид 2-3. Задачи, решение которых описывается рекурсивным алгоритмом. Рекурсивное описание конструкций из бесконечного числа элементов.

Тип 13. Задачи на создание геометрических конструкций с использованием рекурсивных описаний.

Пример задачи типа 13. Разложением квадрата называется разбиение его на конечное число прямоугольников, стороны которых параллельны сторонам квадрата. Разложение называется примитивным, если оно не является разбиением более крупного разложения. При каких *n* существует примитивное разложение квадрата на *n* прямоугольников. (Для *n* > 6 построить рекурсивное описание разбиения.)

Замечания по реализации. Решение должно описываться рекурсивным алгоритмом. Интерфейс геометрической рекурсии можно сделать аналогичным интерфейсу скриптов в «The Geometer's Sketchpad — 2».

Параметризация. В этом сюжете прямая параметризация неочевидна. Можно поэкспериментировать с фигурами, составленными из прямоугольников. Возможно, это приведет к камуфлирующей параметризации.

Замечания по верификации ответа. Структура интерфейса ограничивает все решения только разбиением на прямоугольники со сторонами, параллельными сторонам квадрата. Проверять нужно только примитивность разбиения для различных значений *n*. Это, на первый взгляд, отнюдь не простая верификация.

Вид 3. Задачи на геометрическое конструирование.

Подвид 3-1. Задачи на конструирование геометрического инварианта.

Тип 14. Задачи на конструирование по заданному множеству подмножества, инвариантного относительно вариации некоторых параметров исходного множества.

Пример задачи типа 14. В лесу, имеющем форму выпуклого многоугольника площадью 1 кв. км, заблудился человек. Докажите, что он всегда может выйти из леса, пройдя путь, меньший 2507 м.

Указание. Для доказательства укажите траекторию движения, которая всегда приведет к выходу из леса независимо от его формы.

Замечания по реализации. Решением является полуокружность радиуса 2507/π. Таким образом, для задания траектории удобен геометрический ввод с набором примитивов для описания кривых на плоскости.

Параметризация. Камуфлирующая параметризация возможна изменением площади леса и соответственно длины пути. Содержательная параметризация может быть связана с описанием параметрического семейства областей, формы которых может иметь лес. Это может привести к изменению стратегии выхода и соответственно к изменению длины кратчайшего пути. Для подтверждения достаточно рассмотреть лес в форме круга, прохождение которого можно сделать по любой прямой и расстояние при этом будет меньше, чем в задаче.

Замечания по верификации ответа. Траектория проверяется по двум признакам:

1) не превышает ли ее длина заданного значения;
2) в рзличных примерах областей — содержится ли она в этой области.

Подвид 3-2. Задачи на конструирование оптимального геометрического решения.

Тип 15. Задачи на нахождение оптимального расположения геометрических фигур.

Пример задачи типа 15. В квадрат площади 5 поместить 9 многоугольников каждый площади 1 так, чтобы наибольшая площадь пересечения двух многоугольников была наименьшей.

Замечания по реализации. Среда реализации аналогична «The Geometer's Sketchpad». Произвольные многоугольники площади 1 генерируются по числу сторон. Форма их меняется мышкой с сохранением площади. Далее построенные многоугольники как твердые тела располагаются в заданном квадрате.

Параметризация. Можно менять параметры исходной области, ее площадь, число размещаемых фигур и их площадь. Можно добавлять ограничения на размещаемые фигуры (выпуклые, с заданным числом вершин, равные и пр.).

Замечания по верификации ответа. Проверяется попадание фигур внутрь заданной области. Вычисляется наибольшая площадь пересечения, которая сравнивается с эталонной (либо с той, которая получается случайным разбросом фигур внутри области). Если достигается эпсилон-равенство, задача принимается. Возможно, на последнем этапе следует запускать скрытую локальную оптимизацию и, если расположение фигур при этом принципиально не меняется, что можно проверить отношением зацепления между фигурами, то ответом считается минимакс после локальной оптимизации решения.

Архитектура системы

Автоматизированная система проведения научных соревнований должна обеспечивать выполнение следующих требований:

1) поддерживать продуктивную деятельность участников, решение творческих задач;

2) обеспечить преемственность с традиционной системой соревнований, использовать методическое богатство наработанных форм проведения научных соревнований;

3) разделить предметную сферу деятельности ученика (процесс взаимодействия со средой задачи) и дидактическую составляющую среды, связанную с организацией процесса проведения научных соревнований и подготовки к ним [1];

4) обеспечить автоматизацию процессов регистрации, диалога, проверки, регламент соревнований, подготовку к олимпиаде;

5) система должна удовлетворять современным требованиям, предъявляемым к информационно-коммуникационным системам: быть платформонезависимой, иметь web-интерфейс и пр.

Для выполнения этих условий нами предложена архитектура системы со следующими свойствами:

1) система включает в себя два связанных блока, относящихся к работе с задачами: один блок поддерживает взаимодействие ученика с задачей, реализуя интерфейс, специфичный для данного типа задач; другой отвечает за обработку решения, полученного в процессе взаимодействия ученика со средой задачи; выделение этих блоков в отдельную подсистему позволяет сделать систему открытой, постепенно наращивая спектр типов задач, используемых в дистанционных конкурсах, тем самым позволяя соединять в одном соревновании тестовые задания с простой формой ответа с задачами со сложным интерфейсом;

2) система включает в себя блок управления соревнованием, в котором формируется регламент участника и исполнение которого поддерживается системой автоматически; в этом блоке осуществляется администрирование банков задач и соревнований, управление правами на участие в соревнованиях, выдача

участникам условий задач и модулей для поддержки решения задач, слежение за временем, передача решений участников модулям проверки решений, составление таблицы результатов, сохранение истории отосланных решений, публикация результатов соревнования в соответствии с правами на просмотр.

Заключение

Интеграция математических задач в систему автоматизированной поддержки научных соревнований предъявляет требования как к структуре и подготовке олимпиадных задач, так и к архитектуре самой системы.

Подготовка математических задач предполагает точное описание операционной среды задачи, в которой может быть представлено любое ее решение и механизмы верификации решения. Большое значение имеет параметризация условий задач, позволяющая варьировать задачи при сохранении их содержания и уровня сложности.

Архитектура системы должна обеспечивать реализацию этих принципов и содержать два базовых блока: блок для поддержки необходимой операционной среды задачи и диалога с обучаемым (входит в состав клиентской части системы); блок для верификации решения задачи (входит в состав серверной части системы).

Отдельный блок должен обеспечивать поддержку регламента соревнования и всех бизнес-процессов, относящихся к проведению научного соревнования.

СПИСОК ЛИТЕРАТУРЫ

1. *Башмаков М. И., Поздняков С. Н., Резник Н. А.* Информационная среда обучения. СПб.: Свет, 1997.

2. *Богданов М. С.* Автоматизация проверки решения задачи по формальному описанию ее условия // Компьютерные инструменты в образовании. 2006. № 4. С. 51–57.

3. Веб-судоку как пример конструирования общего информационного пространства // Компьютерные инструменты в образовании». 2005. № 6.

4. *Иванов С. Г.* Компьютерная поддержка решения математических задач как способ организации продуктивной деятельности учащихся: Дис. … канд. пед. наук. М., 2004.

5. *Манцеров Д. И.* Среда KD-Verifier 2: верификация решений задач по математике // Компьютерные инструменты в образовании. 2006. № 4.

6. *Поздняков С. Н.* Конструирование общего информационного пространства // Материалы IX Междун. конфер.-выставки. ИТО–99: 9–12 ноября 1999 г. М., 1999.

7. *Поздняков С. Н.* Моделирование информационной среды как технологическая основа обучения математике: Дис. … д-ра пед. наук. М., 1999.

8. *Посов И. А.* Автоматическая генерация задач // Компьютерные инструменты в образовании. 2007. № 1. С. 54–62.

9. *Рукшин С. Е.* Технологическая поддержка стимулирующих занятий математикой // Компьютерные инструменты в школе. 2009. № 6. С. 3–12.

10. *Степанов А. В.* Система компьютерной генерации заданий по математике // Компьютерные инструменты в образовании. 2000. № 3–4. С. 28–31.

11. *Bannon L., Budker S.* Constructing Common Information Spaces, 1997.

12. *Bogdanov M., Pozdnyakov S., Pukhov A.* Multiplicity of the knowledge representation forms as a base of using a computer for the studying of the discrete mathematics // The 9th International conference «Teaching Mathematics: Retrospective and Perspectives». Vilnius Pedagogical University, 16–17 May 2008.

13. *Freiman V., Kadijevich D., Kuntz G., Pozdnyakov S., Stedoy I.* Challenging mathematics beyond the classroom enhanced by technological environments // Challenging Mathematics In and Beyond the Classroom: Chapter 3. Springer Science + Business Media, LLC 2009.
14. *Nason R., Woodruff E.* Online Collaborative learning in Mathematics: Some Necessary Innovations // T. S. Roberts. Online Collaborative Learning: Theory and Practice. Information Science Publishing. 2004. P. 103–131.

REFERENCES

1. *Bashmakov M. I., Pozdnjakov S. N., Reznik N. A.* Informacionnaja sreda obuchenija. SPb.: Svet, 1997.
2. *Bogdanov M. S.* Avtomatizacija proverki reshenija zadachi po formal'nomu opisaniju ee uslovija // Komp'juternye instrumenty v obrazovanii. 2006. № 4. S. 51–57.
3. Veb-sudoku kak primer konstruirovanija obwego informacionnogo prostranstva // Komp'juternye instrumenty v obrazovanii. 2005. № 6.
4. *Ivanov S. G.* Komp'juternaja podderzhka reshenija matematicheskih zadach kak sposob organizacii produktivnoj dejatel'nosti uchawihsja: Dis. … kand. ped. nauk. M., 2004.
5. *Mancerov D. I.* Sreda KD-Verifier 2: verifikacija reshenij zadach po matematike // Komp'juternye instrumenty v obrazovanii. 2006. № 4.
6. *Pozdnjakov S. N.* Konstruirovanie obwego informacionnogo prostranstva // Materialy IX Mezhdun. konfer.-vystavki. ITO 99: 9–12 nojabrja 1999 g. M., 1999.
7. *Pozdnjakov S. N.* Modelirovanie informacionnoj sredy kak tehnologicheskaja osnova obuchenija matematike: Dis. … d-ra ped. nauk. M., 1999.
8. *Posov I. A.* Avtomaticheskaja generacija zadach // Komp'juternye instrumenty v obrazovanii. 2007. № 1. S. 54–62.
9. *Rukshin S. E.* Tehnologicheskaja podderzhka stimulirujuwih zanjatij matematikoj // Komp'juternye instrumenty v shkole. 2009. № 6. S. 3–12.
10. *Stepanov A. V.* Sistema komp'juternoj generacii zadanij po matematike // Komp'juternye instrumenty v obrazovanii. 2000. № 3–4. S. 28–31.
11. *Bannon L., Budker S.* Constructing Common Information Spaces, 1997.
12. *Bogdanov M., Pozdnyakov S., Pukhov A.* Multiplicity of the knowledge representation forms as a base of using a computer for the studying of the discrete mathematics // The 9th International conference «Teaching Mathematics: Retrospective and Perspectives». Vilnius Pedagogical University, 16–17 May 2008.
13. *Freiman V., Kadijevich D., Kuntz G., Pozdnyakov S., Stedoy I.* Challenging mathematics beyond the classroom enhanced by technological environments // Challenging Mathematics In and Beyond the Classroom: Chapter 3. Springer Science + Business Media, LLC 2009.
14. *Nason R., Woodruff E.* Online Collaborative learning in Mathematics: Some Necessary Innovations // T. S. Roberts. Online Collaborative Learning: Theory and Practice. Information Science Publishing. 2004. P. 103–131.