

Описание элементов полей ввода-вывода в компьютерных программах тестирования

В статье [11] представлен разработанный нами стандарт описания процесса создания на экране монитора окон, посредством которых производится вывод вопросов и ввод ответов во время компьютерного тестирования. Описание указанного процесса согласно предложенному стандарту осуществляется с помощью форматизирующей строки, содержащей: названия элементов и их параметры, а также группирующие и вспомогательные символы. С помощью элементов стандарта описываются объекты программы. Большинство объектов создают окна ввода-вывода на экране монитора. Следующей целью наших исследований являлось – разработать минимальный возможный набор элементов стандарта, который необходим для проведения компьютерного тестирования по математике и информатике.

Выбор элементов стандарта проводился параллельно с созданием использующей их программы и её апробации. Этот подход позволил выбрать достаточно полный набор элементов для тестирования в указанных выше областях знаний. Для предлагаемых элементов подобраны названия, которые, во-первых, указывают на их назначения и, во-вторых, совпадают по названиям с аналогичными по назначению элементами, используемыми в других языках программирования. Для элементов предложены параметры, их названия, назначения и принимаемые ими значения. Принцип подбора названий параметров был такой же, какой и для элементов: названия должны соответствовать их назначениям или совпадать с названиями аналогичных параметров в других языках программирования. Параметры определяют свойства объектов. В частности, если элементы создают окна, то их параметры определяют вид окон (их место расположения,

размеры, оформление) и последовательность создания окон. Все предложенные в статье элементы собраны в таблицу, в столбцах которой указываются названия и назначения элементов. Внутри таблицы элементы построены построчно объединены в четыре группы: элементы ввода, элементы вывода, элементы ввода-вывода, группирующие элементы. Все параметры собраны в другую таблицу, в столбцах которой указываются название, назначение и значение параметров. Внутри этой таблицы указываются целыми строками элементы, для которых предназначены параметры. После каждой таблицы даны необходимые пояснения к отдельным их позициям.

В конце статьи представлен пример использования стандарта при создании окна ввода коэффициентов многочлена. Пример показывает значительную компактность и простоту записи. Кроме того авторами написана программа тестирования на языках программирования HTML, JavaScript, PHP, основанная на предложенных в статье элементах стандарта, которая позволяет проводить тестирование по математике и информатике. Программа размещена на сайте [20]. По этой программе многократно проводилось тестирование студентов НИУ Московского государственного университета.

Предложенный состав элементов и их параметров удобен в использовании и не требует высокого уровня знания языков программирования создателями тестовых задач.

Ключевые слова: компьютерное тестирование, стандарт, JSON, объект, свойства объекта, параметры элемента, форматизирующая строка, поле ввода-вывода.

Igor V. Loshkov¹, Dmitry I. Loshkov²

¹Moscow State University of Civil Engineering (National Research University), Moscow, Russia

²University of Maine, Orono, Maine, USA

Describing of elements IO field in a testing computer program

A standard of describing the process of displaying interactive windows on a computer monitor, through which an output of questions and input of answers are implemented during computer testing, is presented in the article [11]. According to the proposed standard, the description of the process mentioned above is performed with a format line, containing element names, their parameters as well as grouping and auxiliary symbols. Program objects are described using elements of standard. The majority of objects create input and output windows on a computer monitor. The aim of our research was to develop a minimum possible set of elements of standard to perform mathematical and computer science testing.

The choice of elements of the standard was conducted in parallel with the development and testing of the program that uses them. This approach made it possible to choose a sufficiently complete set of elements for testing in fields of study mentioned above. For the proposed elements, names were selected in such a way: firstly, they indicate their function and secondly, they coincide with the names of elements in other programming languages that are similar by function. Parameters, their names, their assignments and accepted values are proposed for the elements. The principle of name selection for the parameters was the same as for elements of the standard: the names should correspond to their assignments or coincide with names of similar parameters in other programming languages. The parameters define properties of objects. Particularly, while the elements of standard create windows, the parameters define object properties (location,

size, appearance) and the sequence in which windows are created. All elements of standard, proposed in this article are composed in a table, the columns of which have names and functions of these elements. Inside the table, the elements of standard are grouped by row into four sets: input elements, output elements, input-output elements, grouping elements. All parameters are collected into another table, columns of which have names, assignments and values of parameters. Elements, for which the parameters are intended, are indicated in whole lines inside of this table. After each table, the necessary explanations are given to some of its items.

An example of use of the standard for creating an input window of polynomial coefficients is demonstrated at the end of this article. The example shows significant compactness and ease of recording. Moreover, the testing program, based on the elements of standard, proposed in this article was written on HTML, JavaScript, PHP program languages and allows to perform mathematical and computer science testing. The program is available on the website [20]. Students of Moscow State (National Research) University of Civil Engineering were tested frequently using this program.

The composition of elements and their parameters proposed above is convenient and does not require the creators of test problems to have a high level knowledge of programming languages.

Keywords: computer testing, standard, JSON, object, object properties, parameters of the element, formatting string, IO field.

1. Введение

Процесс обучения состоит из двух главных компонент: вначале происходит передача обучающимся знаний, умений и навыков, а затем их закрепление и проверка. Если первая компонента обучения может быть достаточно эффективно алгоритмизирована, то вторая — пока не позволяет это вполне хорошо осуществить. Один из путей решения алгоритмизации второй компоненты обучения — создание программ тестирования. Однако проверка знаний преподавателем и программой тестирования существенно отличаются. Преподаватель постоянно находится в диалоге с экзаменуемым, и потому характер вопросов и заданий преподавателя зависит от полученных им ответов. Большинство же программ тестирования реализует простой алгоритм проверки знаний: ставится вопрос — нужно дать ответ. Обзор некоторых таких программ дан в работах [1,2].

Создание каждой тестовой программы — трудоёмкий процесс. Поэтому важно стандартизировать форму хранения вопросов, чтобы была возможность обмена тестовыми вопросами между разработчиками программ. С этой целью компанией Question Mark Computing Ltd был разработан язык представления вопросов QML (Question Mark-up Language) [3], основанный на языке разметки XML [4], широко используемом в среде программистов. Дальнейшее развитие стандартизация представления вопросов получила в разработанной IMS Global Learning Consortium спецификации IMS Questions & Tests Interoperability Specification [5]. Затем этим консорциумом были разработаны спецификации: IMS Content Packaging — для методов хранения, обмена, управления модулями обучающих программ [6]; IMS Simple Sequencing — для управления

потоком контента между обучающимся и программой [7]. В дальнейшем в США по правительственной программе ADL (Advanced Distributed Learning) была осуществлена интеграция различных стандартов и спецификаций (в том числе упомянутых IMS) в единую модель контента — спецификацию SCORM [8,9], позволяющую совместно использовать объекты в распределенной обучающей среде. На данный момент SCORM, фактически стала международным стандартом предоставления электронного образовательного контента. На основе этой спецификации в частности российской командой программистов компании «ГиперМетод» разработана программная платформа eLearning Server 4G [10], получившая широкое распространение не только в России, но и за рубежом.

В настоящий момент начинает развиваться более сложный подход к тестированию: ответ на поставленный вопрос должен происходить в диалоге тестирующегося и компьютерной программы, но в жёстко алгоритмизированных рамках. Принципы и методы такого тестирования в настоящее время ещё не разработаны. В рамках этого подхода компьютерное тестирование может быть направлено, например, на создание задач, в которых тестирующийся должен выбирать варианты типов ответов, или в течение решения одной задачи требуется введение промежуточных ответов поэтапно. В этом примере уже происходит активное взаимодействие человека с программой, которое осуществляется с помощью экрана монитора. С использованием спецификации SCORM можно создавать программы, активно взаимодействующие с тестирующимся. Однако в этом случае уже нельзя будет использовать имеющиеся интерфейсы разработанных программ и новый тип вопросов надо писать на языке указанных выше спе-

цификаций (созданном на основе XML). И если мы хотим, чтобы преподаватели (не обладающие большими навыками в программировании) сами составляли задачи для тестов, надо предложить им возможно более простой и наглядный стандарт для написания тестов. Это было нами сделано в работе [11]. Предложенный нами стандарт описания поля ввода-вывода (СОПВВ) предлагает производить описание поля ввода-вывода (области на экране монитора, на которой происходит взаимодействие человека с программой ЭВМ) с помощью формирующей текстовой строки. Главной компонентой формирующей строки является элемент, определяющий объект программы, который создаёт окно на поле ввода-вывода.

Теперь в рамках предложенного стандарта необходимо составить перечень элементов и их параметров и свойств, необходимых для составления тестов. Состав элементов СОПВВ будет зависеть от области знаний, для которой проводится тестирование. В настоящей статье мы поставили цель — предложить минимальный возможный перечень элементов для тестирования по математике и информатике, их наименования, назначения и параметры, а также наименования и назначения параметров. Наименования элементов мы старались подбирать максимально приближенными к наименованиям соответствующих элементов в стандарте как QML, так и HTML [12]. Так как цель настоящей статьи была предложить минимальный набор элементов, то возможно дальнейшее их расширение.

2. Список элементов

Рассмотрим возможный состав элементов СОПВВ.

Согласно [11], элементы можно объединить в следующие группы:

Таблица 1

Название элемента	Назначение элемента
<i>Элементы вывода</i>	
"formtext" ⁽¹⁾	вывод текста, который будет отображаться (форматироваться) в соответствии со стандартом используемой программы для тестирования
"text" ⁽²⁾	вывод текста, который будет отображаться «как есть» – без изменения программой тестирования
"formula"	вывод в научном виде формулы, записанной в каком-либо указанном стандарте записи формул с помощью текстовых строк
"image"	вывод графического файла
"hidden"	передача невидимого набора данных
<i>Элементы ввода</i>	
"number"	ввод числа
"string"	ввод однострочного текста
"textarea"	ввод многострочного текста
"sqrt" ⁽³⁾	ввод числа под знаком корня
"vector" ⁽⁴⁾	ввод вектора
"array" ⁽⁴⁾	ввод матрицы
"polynomial" ⁽⁵⁾	ввод числовых сомножителей (коэффициентов) для слагаемых многочлена
"informula" ⁽⁶⁾	ввод формулы в каком-либо указанном стандарте разметки формул с помощью текстовых строк
"exec"	ввод исполняемого кода
<i>Элементы ввода-вывода</i>	
"select" ⁽⁷⁾	выбор одного или нескольких ответов из выпадающего списка с вариантами ответов или управление одним или несколькими элементами на основании выпадающего списка
"checkbox" ⁽⁷⁾	выбор одного или нескольких ответов из нескольких заданных вариантов ответа или управление одним или несколькими элементами на основании выбранных вариантов ответа
"radio" ⁽⁷⁾	выбор одного ответа из нескольких заданных вариантов или управление одним или несколькими элементами на основании выбранного варианта ответа
"gallery"	выбор ответа из галереи графических файлов или управление одним или несколькими элементами на основании выбранных вариантов ответа
<i>Управляющие элементы</i>	
"button"	кнопка для управления другими элементами
"div"	резервирование места для вставки в это место окна, создаваемого другим элементом

– группа элементов вывода;
 – группа элементов ввода;
 – группа элементов ввода-вывода;
 – группа управляющих элементов.

Все элементы имеют две части: название и список параметров. Название обрамляется в кавычки. В табл. 1 представлен возможный список элементов, разбитый на группы с указанием назначения каждого элемента.

Укажем особенности применения некоторых элементов.

⁽¹⁾ Например, если для тестирования используется Интернет-браузер, то с помощью элемента «formtext» может выводиться текст, отформатированный с помощью html-тэгов, содержащихся в самом тексте.

⁽²⁾ Если же мы будем использовать элемент «text», то текст должен изображаться со всеми тэгами без форматирования. Таким образом, результат вывода текста с помощью элемента «formtext» может зависеть от возможностей программы тестирования. Вывод же текста с помощью элемента «text» должен быть идентичен во всех программах.

⁽³⁾ Целесообразность введения элемента «sqrt» связана с частым использованием квадратного корня в математических формулах.

⁽⁴⁾ Структурно окна элементов «vector» и «array» есть упорядоченная последовательность окон элемента «number». Вектора можно вводить, также используя одномерные матри-

цы, то есть, используя элемент «array» с соответствующими параметрами, оставляющими у матрицы либо одну строку, либо один столбец.

⁽⁵⁾ Целесообразность введения элемента «polynomial» также связана с частым использованием в математических формулах полиномов. Использование этого элемента сокращает запись формирующей строки. Структурно окно элемента «polynomial» – последовательность пар окон элементов «number» (ввод числового сомножителя для слагаемого) и «formtext» (отображение остальной части слагаемого), и вставленных между парами окон знаков сложения +.

⁽⁶⁾ Элемент «informula» должен позволить вводить фор-

мулы с помощью текстовых строк, использующих какой-либо язык ввода формул, например MathML [13], TeX [14,15]. При этом для визуального контроля правильности ввода формулы целесообразно результат ввода текста в окно элемента «informula» сразу отображать в окне с помощью элемента «formula».

⁽⁷⁾ Элементы «select», «checkbox», «radio» создают окна аналогичные полям со списками, создаваемым соответствующими тэгами стандарта HTML [12].

3. Параметры элементов

Каждый элемент имеет набор параметров. В [11] рассмотрены правила описания

параметров. В частности, параметры элементов в формирующей строке задаются с помощью текста следующей структуры.

$$A_n : \{B_1 : C_1, B_2 : C_2, B_3 : C_3, \dots, B_k : C_k\}.$$

Здесь A_n – название элемента или блок элементов, $B_1, B_2, B_3, \dots, B_k$ – название параметра, обрамлённое в кавычки, а $C_1, C_2, C_3, \dots, C_k$ – значение параметра, обрамлённое в кавычки (цифры и логические значения могут не обрамляться кавычками).

В таблице 2 представлен возможный список параметров с указанием назначения каждого параметра и его значения для элементов, перечисленных в табл. 2.

Пояснения к таблице списка параметров

Значения параметров можно вводить в формирующую строку в процессе работы программы посредством использования внедряемого текста (см. статью [11]), а именно путём замены конструкции `/# <переменная> #/` в формирующей строке на текст. При этом переменной с именем `<переменная>` должно быть присвоено программой текстовое значение до использования формирующей строки. В любом месте формирующей строки можно вставлять комментарии, обрамлённые слева парой символов `/*`, а справа – `*/` (см. статью [11]). Таким образом, в значениях параметров пары символов: `/#`, `*/` и `/*`, `*/` могут трактоваться в том

Таблица 2

Название параметра	Назначение параметра	Значение параметра
Параметры, общие для всех элементов или блоков		
"meta"	задать общие настройки блока обработки элементов поля ввода-вывода	аналогичны атрибутам тэга meta стандарта HTML [16]; по умолчанию значения передаются от блока формирования вопросов программы тестирования
"ident"	задать идентификатор элемента ⁽⁸⁾	любой набор символов; каждый элемент должен иметь уникальный идентификатор
"allocation"	задать порядок размещения окон на поле ввода-вывода ⁽⁹⁾	"inserted" "linewise" "column" – вложенное построчное постолбцовое размещение; по умолчанию – вложенное размещение
"style"	задать стиль выводимых символов	задаётся по правилам стандарта каскадной таблицы стилей CSS ⁽¹⁰⁾
"cell" ⁽¹¹⁾	задать стиль ячейки поля ввода-вывода, в которой расположено окно элемента	задаётся по правилам стандарта каскадной таблицы стилей CSS ⁽¹⁰⁾
"count" ⁽¹²⁾	задать количество повторений окон	Число "auto" "<имя переменной>", здесь <имя переменной> – имя переменной, записанное с помощью параметра "saved" элемента ввода-вывода
"orientation" ⁽¹²⁾	задать ориентацию расположения окон при их повторении	"vertical" "horizontal" – вертикальное горизонтальное расположение, по умолчанию – горизонтальное расположение
"collection" ⁽¹³⁾	задать числа, сопоставляемые окнам при их повторении	число или набор чисел
"numerate" ⁽¹⁴⁾	задать место вставки числа в выводимый текст, если есть повторение окон	любой допустимый набор символов
"index" ⁽¹²⁾	определить расположение числа, определяемого параметром "numerate", относительно уровня строки	"top" – выше, "bottom" – ниже; по умолчанию число не сдвигается
"attribute"	ввести атрибуты элемента или блока	перечень атрибутов, разделённый пробелами и обрамлённый в кавычки
"choice"	сохранить выбор при значении параметра "divarication" равным "different"	равняется индексу параметра "variants", соответствующему выбранному варианту ответа для элемента ввода-вывода

Название параметра	Назначение параметра	Значение параметра
Параметры, общие для всех элементов		
"line"	провести линию	"top" "bottom" "left" "right" – линия сверху снизу слева справа окна
"source" ⁽¹⁵⁾	указать имя переменной, из которой надо прочитать данные и указать что окно первоначально невидимо или скрыто (является шаблоном окна)	набор символов без пробелов: буквы, цифры, подчеркивание
"target" ⁽¹⁵⁾	задать имя окна, которое замещается данным окном	"self" допустимый набор символов в кавычках; такой же набор символов должен быть для значения параметра "container" окна, которое замещается
"container" ⁽¹⁵⁾	задать имя окна, как то которое может замещаться другим окном; этот параметр, обязателен для элемента "div"	допустимый набор символов в кавычках, такой же набор символов должен быть для значения параметра "target" окна, которое замещает
Параметры, общие для всех элементов ввода-вывода		
"divarication" ⁽¹⁶⁾	определить разные, либо одинаковые иницируются окна для разных вариантов ответов	"different" "same" – разные одинаковые, по умолчанию – одинаковые
"saved"	задать имя переменной, в которой будет сохраняться выбранный вариант ответа	набор символов без пробелов: буквы, цифры, подчеркивание
Параметры, общие для всех управляющих элементов		
"insert" ⁽¹⁴⁾	задать имя переменной, в которой будут сохраняться данные	набор символов без пробелов: буквы, цифры, подчеркивание
Параметры элемента "formtext"		
"value"	задать выводимый текст	любой допустимый набор символов
"type" ⁽¹⁷⁾	указать язык программирования	название языка программирования, по умолчанию – язык HTML
"encoding"	указать кодировку символов текста	используемая кодировка текста, по умолчанию – Windows-1251 (cp1251) [17] или задаётся в настройках
Параметры элемента "text"		
"value"	задать выводимый текст	любой допустимый набор символов
"encoding"	указать кодировку символов текста	используемая кодировка текста, по умолчанию – Windows-1251 (cp1251) или задаётся в настройках
Параметры элемента "formula" ⁽¹⁸⁾		
"value"	задать текст формулы	текстовая строка, в которой записана формула с символами форматирования языка разметки формул
"type"	задать язык разметки формул, переводящий текстовую строку, в которой записана формула, в научный вид	название языка разметки формул
Параметры элемента "image"		
"src"	ввести путь к графическому файлу	URL-адрес нахождения графического файла
Параметры элемента "hidden"		
"value"	задать невидимые данные	задаются по стандарту JSON [18]
Параметры элемента "number"		
"size"	задать длину окна	положительное целое число; по умолчанию – значение 3
Параметры элемента "string"		
"size"	задать длину окна	положительное целое число; по умолчанию – значение 8
Параметры элемента "informula" ⁽¹⁸⁾		
"size"	задать длину окна	положительное целое число; по умолчанию – значение 8
"type"	задать язык разметки формул, переводящий текстовую строку, в которой записана формула, в научный вид	название языка разметки формул
"function" ⁽¹⁹⁾	задать функцию, которая использует полученные невидимые данные для проверки правильности ввода формулы	имя используемой функции и в скобках имя элемента "hidden", из которого берутся параметры функции

Название параметра	Назначение параметра	Значение параметра
Параметры элемента "exec"		
"value"	ввести программный код	любой допустимый набор символов для используемой программы
Параметры элемента "textarea"		
"row"	задать количество строк окна	целое число; по умолчанию – значение 2
"col"	задать количество столбцов окна	целое число; по умолчанию – значение 80
"marking" ⁽²⁰⁾	разметить каждую строку по позициям	"yes" – есть разметка; по умолчанию разметки нет
Параметры элемента "sqrt"		
"size"	задать длину окна под знаком корня	положительное целое число; по умолчанию – значение 3
Параметры элемента "vector" ⁽²¹⁾		
"ncol"	задать количество компонент вектора и определяет, что вектор является строкой	положительное число
"nrow"	задать количество компонент вектора и определяет, что вектор является столбцом	положительное число
"size"	задать длину поля для каждой компоненты	положительное целое число; по умолчанию – значение 3
"bracket"	задаёт форму скобок справа и слева по краю окна	"round" "square" "figure" "right" – круглая квадратная фигурная прямая скобки; по умолчанию – круглые скобки
Параметры элемента "array" ⁽²¹⁾		
"ncol"	количество столбцов матрицы	положительное число
"nrow"	количество строк матрицы	положительное число
"size"	задать длину поля для каждой компоненты	положительное целое число; по умолчанию – значение 3
"bracket"	задаёт форму скобок справа и слева по краю окна	"round" "square" "figure" "right" – круглая квадратная фигурная прямая скобки; по умолчанию – круглые скобки
Параметры элемента ввода "polynomial"		
"count"	задать количество слагаемых многочлена	число "auto" "<имя переменной>", здесь <имя переменной> – имя переменной, записанное с помощью параметра "saved" элемента ввода-вывода
"collection"	задать числа, которые могут использоваться как индексы для слагаемых многочлена	число или набор чисел ⁽¹³⁾
"char"	определить строку, входящую в каждое слагаемое многочлена	набор символов ⁽²²⁾
"index"	определить вертикальное расположение индекса для строки, заданной параметром "char"	"top" –вверху, "bottom" –внизу, по умолчанию индекс вставляется на уровне строки
"size"	определить длину окна ввода численного коэффициента в каждом слагаемом многочлена	положительное целое число; по умолчанию – значение 3
"zero"	изменить вид слагаемого при значении индекса равном нулю	"no" – слагаемое отсутствует, "yes" – слагаемое без индекса; по умолчанию – слагаемое с индексом нуль
"one"	изменить вид слагаемого при значении индекса равном единице	"no" – не вставлять индекс, "yes" – вставлять индекс; по умолчанию индекс вставляется
Параметры элементов с множественностью выбора ⁽²³⁾		
"variants"	задать строки текста для соответствующих компонент элемента	задаётся по стандарту JSON [18]
"order"	задать порядок вывода компонент элемента и строк текста	"random" <числа> – случайный порядок номера вопросов, для которых не изменяется порядок вывода; по умолчанию – вывод в порядке задания строк текста
Параметры элемента "select"		
"size"	задать количество одновременно отображаемых строк списка	положительное целое число; по умолчанию – значение 1

Название параметра	Назначение параметра	Значение параметра
Параметры элементов "checkbox", "gallery", "radio"		
"orient"	задать ориентацию расположения компонент элемента	"vertical" "horizontal" – вертикальное горизонтальное расположение компонент; по умолчанию – вертикальное
"position"	задать место расположения текста относительно компонент элемента	"right" "left" "top" "bottom" – текст расположен справа слева сверху снизу от компонент; по умолчанию текст расположен слева от компонент
Параметры элемента "gallery"		
"url"	ввести путь к графическим файлам без имени самих файлов	URL-адрес нахождения графических файлов без имени файлов, по умолчанию адрес совпадает с местом нахождения программы
"names"	ввести имена графических файлов	имена графических файлов задаются в кавычках через запятую, а весь список обрамляется квадратными скобками
Параметры элемента "button"		
"value"	размещение текста на кнопке	любой допустимый набор символов

смысле, как указано в настоящем абзаце, и потому их нельзя использовать для написания значений параметров.

Символ | в значении параметров заменяет разделительный союз «либо».

⁽⁸⁾ Если не используется параметр `ident`, то программа должна присваивать всем элементам внутренние идентификаторы, которые должны задаваться последовательно в порядке создания элемента.

⁽⁹⁾ Возможные варианты порядка размещения окон на поле ввода-вывода рассмотрены в статье [3].

⁽¹⁰⁾ По правилам стандарта каскадной таблицы стилей CSS [19] свойство и его значение записываются разделёнными двоеточием, пары «свойство:значение» разделяются друг от друга точкой с запятой, вся строка свойств со значениями обрамляется в кавычки.

⁽¹¹⁾ Для размещения окон элементов в необходимой последовательности необходимо поле ввода-вывода разбить на ячейки (области) и затем в эти ячейки поместить окна. Параметр `cell` элемента может использоваться для управления размещением окна в ячейке и видом ячейки (например, указать параметры её границ – форма, цвет, толщина).

⁽¹²⁾ Параметры «count», «orientation», «collection», «numerate», «index» управляют выводом повторяющихся окон. Параметр «count» определяет количество повторений окон. Он имеет значением или число, или слово «auto», или имя переменной, в которой сохраняется число. Параметр «count» может иметь значение «auto», если перед обращением к нему тестирующийся определил только одно число, например, с помощью элемента «number» или с помощью элемента, имеющего параметр «variants», выбрав вариант ответа, задающего одно число. Если же к моменту обращения к параметру «count» тестирующийся не ввёл ещё числа или в процессе ответов может вводиться несколько чисел, то в переменную, из которой берётся число, его запись производится с помощью элемента ввода-вывода, задающего эту переменную с помощью параметра «insert». Ясно, что если к моменту обращения к переменной, указанной параметром «count» в ней число ещё не сохранено, то элемент с соответствующим параметром «count» должен быть скрытым (то есть окно создаётся из шаблона).

⁽¹³⁾ Параметр «collection» позволяет сопоставить число каждому последовательно выводимому окну согласно набору

целых чисел, указанному в значении параметра. Набор значений для параметра «collection» может задаваться перечислением через запятую чисел или массивов чисел, заданных по схеме – $n1:n2:n3$, где $n1$ – первое число массива, $n2$ – шаг изменения последовательности чисел в массиве, $n3$ – граница массива чисел (число из данного массива не может превысить эту границу). В случае отсутствия числа $n2$ шаг изменения массива чисел принимается равным единице. Если определённый числами или массивами набор чисел меньше необходимого количества, то он расширяется по такому правилу: а) следующее число увеличивается на единицу от предыдущего числа, б) следующее число уменьшается на единицу, если после набора чисел стоит символ минус «-». Если перед набором чисел стоит запятая, то первое число набора чисел не определено (индекс в первом слагаемом не ставится), однако оно учитывается при общем подсчёте количества чисел, которое должно быть равно параметру «count». Если количество набора чисел превышает параметр «count», то берутся первые «count» чисел.

⁽¹⁴⁾ Параметр «numerate» определяет последовательность символов в выводимом тексте, которая заменяется сопоставляемым окну числом.

⁽¹⁵⁾ Элемент, имеющий параметр «source», но не имеющий параметра «target», создаёт невидимое окно, которое занимает тоже место, которое бы занимало видимое окно. Если же этот элемент имеет одновременно параметр «target», то первоначально на поле ввода-вывода не создаётся никакого окна (так называемое, скрытое окно или шаблон окна). Управляющий элемент, содержащий параметр «insert», воздействует на подчинённый ему элемент, содержащий параметр «source», значение которого совпадает со значением «insert». Этот управляющий элемент (с помощью его окна) делает окно подчинённого элемента видимым (если оно вначале было не видимым), либо на основании подчинённого элемента создаёт из шаблона окно и помещает его на место того окна, которое создано другим элементом, имеющим значение параметра «container» такое же как и значение параметра «target» подчинённого элемента. Если значение параметра «target» равно «self», то подчинённый элемент создаёт окно на месте, которое определяется местом расположения подчинённого элемента в формирующей строке (то есть скрытое окно становится не скрытым).

⁽¹⁶⁾ Параметр «divarication» позволяет указать будет ли шаблон элемента изменяться или нет.

⁽¹⁷⁾ Язык программирования необходимо указывать для правильной интерпретации формирующих символов, встречающихся в тексте. Таким образом, программа тестирования должна уметь интерпретировать разные языки форматирования текста.

⁽¹⁸⁾ Если происходит набор формулы в виде текстовой строки в окне элемента

Введите коэффициенты: sin(kx)+ sin(3kx)+ sin(5kx)

Рис. Окно ввода-вывода для примера

«informula», то этот набор можно отображать в научном виде в окне элемента «formula». С этой целью для элемента «informula» необходимо использовать параметр «target», а для элемента «formula» – параметр «container». При этом значения двух параметров должны совпадать.

⁽¹⁹⁾ Программная «function» должна сопоставлять набору невидимых данных набор новых данных для оценки правильности ответа тестирующегося.

⁽²⁰⁾ Разметка окна ввода «textarea» необходима в случае, когда ответ зависит также и от позиции, в которой записан символ. Разметку можно осуществлять с помощью, например, засечек. Шрифт ввода ответа должен быть моноширинным.

⁽²¹⁾ Элементы «vector» и «array» для ввода каждого из компоненты-числа создают наборы внутренних подокон, которые аналогичны окну элемента «number», и потому исходные элементы имеют параметр «size», который задаёт размер внутренних подокон.

⁽²²⁾ Если этот набор символов будет содержать строку, указанную в параметре «numerate», то эта строка будет заменяться индексом. В противном случае индекс записывается в конце.

⁽²³⁾ Элементами с множественностью выбора в частности являются: «select», «checkbox», «gallery», «radio», «gallery».

Пример отображения элемента «polynomial»

Написав следующую формирующую строку

```
(
  «formtext»:
  {
    «value»: “Введите коэффициенты: “
    «polynomial”:
```

```
{
  “count”: 3,
  “collection”: “1:2:5”,
  “char”: “sin(nkx)”,
  «numerate»: “n”,
  “one”: “no”
}
```

) мы получим окно в поле ввода-вывода, представленное на рисунке.

4. Заключение

В настоящей статье мы предложили наиболее употребительные элементы формирующей строки, а также элементы, которые можно использовать в тестах по математике и информатике. Для этих элементов мы определили параметры, задаваемые с помощью формирующей строки и определяющие свойства соответствующих объектов программы. При этом мы не рассматривали параметры объектов, не указываемые в формирующей строке и которые должны позволить правильно обработать введённые в окна ответы тестирующихся. Изучение способов анализа ответов тестирующихся составляет самостоятельный предмет.

Авторами разработана программа на языках программирования HTML, JavaScript, PHP, основанная на предложенных в статье элементах стандарта, которая позволяет проводить тестирование по математике и информатике. Программа размещена на сайте [20]. По этой программе многократно проводилось тестирование студентов НИУ Московского государственного университета. Опыт эксплуатации программы показал достаточность указанных элементов для составления широкого класса задач по математике и информатике.

Литература

1. Ананченко И.В. Классификация компьютерных систем тестирования знаний учащихся (научная статья) // Международный журнал экспериментального образования. 2016, № 4–2, 210–213 с. / URL: <http://www.expeducation.ru/ru/article/view?id=9769> (дата обращения: 24.09.2017).
2. ГАУ ДПО Приморский краевой институт развития образования / URL: http://primwiki.ru/index.php/Программы_для_создания_тестов_и_проведения_компьютерного_тестирования
3. Question Mark Computing Ltd / URL: <https://www.questionmark.com/content/qml-question-markup-language>.
4. World Wide Web Consortium / URL: <https://www.w3.org/TR/xml11/>
5. IMS Global Learning Consortium // IMS Questions & Tests Interoperability Specification / URL: <https://www.imsglobal.org/question/index.html>
6. IMS Global Learning Consortium // IMS Content Packaging / URL: https://www.imsglobal.org/content/packaging/cpv1p1p3/imscp_infov1p1p3.html
7. IMS Global Learning Consortium // IMS Simple Sequencing / URL: <https://www.imsglobal.org/simplesequencing/index.html>
8. Advanced Distributed Learning (ADL) / URL: <http://www.adlnet.gov/adl-research/scorm/>
9. Advanced Distributed Learning (ADL) // Sharable Content Object Reference Model (SCORM) 2004. // Перевод с англ. Е.В. Кузьминой. М.: ФГУ ГНИИ ИТТ «Информика», 2005. 29 с. / URL: <http://www.edu.ru/db/portal/e-library/00000053/SCORM-2004.pdf>
10. Компания ГиперМетод (ООО «Ленвэа») / URL: <http://www.hypermethod.ru/product>
11. Лошков И.В., Лошков Д.И. Стандарт описания поля ввода-вывода в программах тестирования (научная статья) // Информатика и образование. 2017, № 5, 60–63 с.
12. World Wide Web Consortium / URL: <https://www.w3.org/TR/html401/index/elements.html>
13. World Wide Web Consortium / URL: <https://www.w3.org/Math/>
14. Кнут Д.Э. Все про TeX = The TeXbook // Пер. с англ. М. В. Лисиной // Протвино: АО RDTeX, 1993. 592 с.
15. Львовский С.М. Набор и верстка в системе LaTeX. 5-е издание. М: МЦНМО, 2014. 398 с.
16. World Wide Web Consortium / URL: <https://www.w3.org/TR/html5/document-metadata.html>
17. URL: <https://ru.wikipedia.org/wiki/Windows-1251>
18. The JSON Data Interchange Standard. 1st Edition. // Standard ECMA-404. October 2013.

References

1. Ananchenko I.V. Klassifikatsiya komp'yuternykh sistem testirovaniya znaniy uchashchikhsya (nauchnaya stat'ya). Mezhdunarodnyy zhurnal eksperimental'nogo obrazovaniya. 2016, No. 4–2, 210–213 p. / URL: <http://www.expeducation.ru/ru/article/view?id=9769> (accessed: 24.09.2017). (In Russ.)
2. GAU DPO Primorskiy kraevoy institut razvitiya obrazovaniya / URL: http://primwiki.ru/index.php/Programmy_dlya_sozdaniya_testov_i_provedeniya_komp'yuternogo_testirovaniya (In Russ.)
3. Question Mark Computing Ltd / URL: <https://www.questionmark.com/content/qml-question-markup-language>. (In Russ.)
4. World Wide Web Consortium / URL: <https://www.w3.org/TR/xml11/>
5. IMS Global Learning Consortium. IMS Questions & Tests Interoperability Specification / URL: <https://www.imsglobal.org/question/index.html>
6. IMS Global Learning Consortium. IMS Content Packaging / URL: https://www.imsglobal.org/content/packaging/cpv1p1p3/imscp_infov1p1p3.html
7. IMS Global Learning Consortium. IMS Simple Sequencing / URL: <https://www.imsglobal.org/simplesequencing/index.html>
8. Advanced Distributed Learning (ADL) / URL: <http://www.adlnet.gov/adl-research/scorm/>
9. Advanced Distributed Learning (ADL). Sharable Content Object Reference Model (SCORM) 2004. Tr. fr. Eng. E.V. Kuz'minoy. M.: FGU GNII ITT «Informika», 2005. 29 p. / URL: <http://www.edu.ru/db/portal/e-library/00000053/SCORM-2004.pdf> (In Russ.)
10. Kompaniya GiperMetod (OOO «Lenvea») / URL: <http://www.hypermethod.ru/product> (In Russ.)
11. Loshkov I.V., Loshkov D.I. Standart opisaniya polya vvoda-vyvoda v programmakh testirovaniya (nauchnaya stat'ya). Informatika i obrazovanie. 2017. No. 5. 60–63 p. (In Russ.)
12. World Wide Web Consortium / URL: <https://www.w3.org/TR/html401/index/elements.html>
13. World Wide Web Consortium / URL: <https://www.w3.org/Math/>
14. Knut D.E. Vse pro TeX = The TeXbook. Tr. fr. Eng. M. V. Lisinoy. Protvino: AO RDTeX, 1993. 592 p. (In Russ.)
15. L'vovskiy S.M. Nabor i verstka v sisteme LaTeX. 5th ed. Moscow: MTsNMO, 2014. 398 p. (In Russ.)
16. World Wide Web Consortium / URL: <https://www.w3.org/TR/html5/document-metadata.html>
17. URL: <https://ru.wikipedia.org/wiki/Windows-1251> (In Russ.)
18. The JSON Data Interchange Standard. 1st Edition. Standard ECMA-404. October 2013.

(Ecma International) / URL: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>

19. World Wide Web Consortium / URL: <https://www.w3.org/Style/CSS/> / Пер. с англ. <https://www.w3.org/Style/CSS/Overview.ru.html>

20 Лошков И.В. / URL: <http://лошков.рф>

(Ecma International) / URL: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>

19. World Wide Web Consortium / URL: <https://www.w3.org/Style/CSS/> / Tr. fr. Eng. <https://www.w3.org/Style/CSS/Overview.ru.html> (In Russ.)

20 Loshkov I.V. / URL: <http://loshkov.rf> (In Russ.)

Сведения об авторах

Игорь Владимирович Лошков

К. ф.-м. н., старший преподаватель,

НИУ МГСУ, Москва, Россия

Эл. почта: loshkovigor@rambler.ru

Тел.: (499) 183 59 94

Дмитрий Игоревич Лошков

Аспирант

Университет штата Мэн, Ороно, Мэн, США

Эл. почта: dmitry.loshkov@maine.edu

Information about the authors

Igor V. Loshkov

Cand. Sci. (Phys.-Math.), Senior Lecturer

Moscow State University of Civil Engineering (National Research University)

E-mail: loshkovigor@rambler.ru

Tel.: (499) 183 59 94

Dmitriy I. Loshkov

Graduate student

University of Maine, Orono, Maine, USA

E-mail: dmitry.loshkov@maine.edu