

Использование шаблонов в виде регулярных выражений в тренировочных и контрольных тестовых вопросах с открытым ответом

Рассматриваются проблемы при автоматизированном тестировании студентов с помощью вопросов в открытой форме, которым соответствует большое количество правильных ответов. Предлагается использовать регулярные выражения, задающие шаблоны ответов. Рассмотрена проблема автоматической выдачи подсказок при прохождении тренировочных тестов. Предлагается использовать генерацию строк, продолжающих частичное совпадение в ответе студента. Это позволяет выдавать подсказки следующего символа или лексемы. Описывается решение, реализованное в виде модуля типа вопроса к популярной системе дистанционного обучения Moodle.

Ключевые слова: электронное обучение, автоматизированное тестирование, регулярные выражения.

USE OF REGULAR EXPRESSIONS AS TEMPLATES IN FORMATIVE AND SUMMATIVE OPEN ANSWER QUESTIONS

The article describes problems in automated students testing using open-form questions with substantial amount of correct answers. Authors propose using of regular expressions as correct answer patterns. The problem of automatic hint generation during training tests is examined as well. To resolve this problem authors propose a matching string generation method, extending partial match from learner's attempt. It is possible to give next correct character and next correct lexeme hints. The paper describes a solution in the form of a question type plugin for the popular e-learning system Moodle.

Keywords: e-learning, automated testing, regular expressions.

Введение

В настоящее время в учебных заведениях многих стран для проверки знаний студентов применяется автоматизированное тестирование с использованием ЭВМ. Одной из наиболее популярных систем дистанционного образования, использующихся для этого, является Moodle [1]. Она используется, наряду с другими российскими вузами, и в ВолгГТУ [2, 3]. В тестах нередко выгодно использовать вопросы с открытым ответом, предлагающие студенту ввести ответ в виде строки. Такие вопросы заставляют студентов думать над заданием тщательнее, чем вопросы с вариантами ответов, способствующие угадыванию. При этом зада-

вать правильные ответы (со стороны преподавателя) в виде перечня строк оказывается недостаточным: зачастую ответ должен допускать синонимы или учитывать множество подряд идущих пробелов как один. В большинстве случаев множество правильных ответов, которое само по себе может оказаться слишком велико для перечисления, удовлетворяет достаточно строгим законам, позволяющим описать его с помощью шаблона. Стандартным инструментом для задания шаблонов совпадения со строками являются регулярные выражения [4].

При обучении языкам значительное время уделяется тренировке составления корректных предложений на них. В языках программирования это могут быть стан-

дартные конструкции – объявление переменных, проверки условий, циклы и т.д. При применении естественных языков такие вопросы могут задаваться как при изучении самого языка, так и для изучения на нем какого-либо предмета.

Когда студент занимается самостоятельной подготовкой и не имеет возможности спросить преподавателя, он может попасть в тупик и потерять время до встречи с преподавателем. Избежать этого ему может помочь автоматически сгенерированная подсказка. Применение автоматических подсказок во время самоподготовки позволяет студентам эффективнее готовиться в любое удобное для них время и меньше зависеть от преподавателя. У преподавателя, в свою очередь,



Олег Александрович Сычев,
к.т.н., доцент кафедры
программного обеспечения
автоматизированных систем
Тел.: (905) 434-53-45
Эл. почта: oasychev@gmail.com
Волгоградский государственный
технический университет
<http://www.vstu.ru>

Oleg A. Sychev,
Candidate of Engineering Science,
Associate Professor, the Automated
Systems Software Department
Tel.: (905) 434-53-45
E-mail: oasychev@gmail.com
Volgograd State Technical University
<http://www.vstu.ru/en>



Валерий Олегович Стрельцов,
магистрант кафедры
программного обеспечения
автоматизированных систем
Тел.: (960) 872-09-59
Эл. почта: vostreltsov@gmail.com
Волгоградский государственный
технический университет
<http://www.vstu.ru>

Valeriy O. Streltsov,
graduate student the Automated Systems
Software Department
Tel.: (960) 872-09-59
E-mail: vostreltsov@gmail.com
Volgograd State Technical University
<http://www.vstu.ru/en>

освобождается время для решения более сложных задач вместо рутинной проверки работ студентов.

Целью данной работы была реализация шаблонных ответов на открытые вопросы при автоматизированном тестировании в СДО Moodle и автоматической генерации подсказок для частично правильных ответов на основе регулярных выражений.

1. Использование регулярных выражений для оценки текстовых ответов

Рассмотрим несколько типовых случаев применения регулярных выражений при оценке ответов на тестовые вопросы в виде строки текста.

Чаще всего бывает желательно обеспечить возможность записи неограниченного числа пробелов между словами, т.е. настроить вопрос таким образом, чтобы наличие лишнего пробела не считалось ошибкой (как это бывает при прямом сопоставлении строк). Это важно для ответов как на естественном, так и на многих формальных (языки программирования, математические формулы) языках. Для этого вместо пробелов используется запись «\s*» или «\s+». Здесь «\s» – это специальное обозначение любого пробельного символа (пробела, табуляции и т.д.). Можно также задать шаблон для цифры («\d»), алфавитно-цифрового символа («\w») и т.д. Символ, следующий после него, обозначает возможное количество повторений (квантификацию): «*» означает любое количество повторений (включая ни одного) и применяется для случаев, когда пробелы могут стоять, но не обязательны, а «+» означает одно или более повторений и применяется в случае, когда наличие хотя бы одного пробела необходимо. Например, регулярное выражение «\w+\s+\w+» совпадет с любыми двумя «словами» (непрерывными последовательностями букв и/или цифр), разделенными одним или более пробелами. Можно также задавать символы через отрицание: «\S» означает любой не-пробельный символ, «\D» означает не-

цифру и «\W» означает не-цифру и не-букву. Если в каком-либо месте может быть вообще любой символ, то для его обозначения используется точка («.»).

Шаблон для одного символа можно также задать через синтаксис квадратных скобок (символьный класс). Например, «[абвгд!?!123]» совпадает с любым из перечисленных в квадратных скобках символов. Подряд идущие символы можно записать еще короче через диапазон: «[а-я]»; диапазоны свободно комбинируются с одиночными символами. Пример из английского языка: назвать животное, чье имя состоит из трех букв, вторая из которых «а». Регулярному выражению «[bcr]at» будут соответствовать «bat», «cat» и «gat». Можно также перечислять все недопустимые в данном месте символы (если это легче), помещая знак «^» после открывающей скобки: «[^а-д1-3!?!]» означает любой символ, кроме тех букв от «а» до «д», цифр от 1 до 3, а также восклицательного и вопросительного знаков.

При составлении регулярных выражений для вопросов на естественных языках (и не только) полезными элементами бывают простые утверждения. Они не совпадают с символами, а проверяют определенные условия и могут прерывать совпадение, если условие неверно.

Простое условие «^» (написанное вне квадратных скобок) совпадает в самом начале строки. Например, регулярное выражение «[а-я]+» найдет совпадение в строке «5 лет» со словом «лет». Однако регулярное выражение «^[а-я]+» его не найдет, поскольку оно обязательно будет начинаться с первого символа строки (ответа обучаемого), а там цифра.

Противоположным простым утверждением является «\$», которое означает конец строки. Регулярное выражение «[а-я]+\$» совпадет только с таким словом, которое является концом строки. Если вы хотите, чтобы конкретное регулярное выражение совпадало целиком с ответом студента (а не с его частью), то его следует взять в скобки, перед скобками поставить «^»,

а после них – «\$» (если вам нужно это для всех регулярных выражений в вопросе, то вы можете также воспользоваться специальной настройкой – см. ниже).

Оставшиеся два простых утверждения – это «\b» и его отрицание «\B». Они обозначают границу и отсутствие границы слова соответственно. Границей слова считается место между алфавитоцифровым символом и символом, не являющимся буквой или цифрой (границей слова также будет наличие буквы или цифры в начале или конце строки). Например, регулярное выражение «\bbeда\b» будет искать совпадение со словом «беда» (а не просто последовательностью букв), поэтому оно найдет совпадение в строке «не беда», но не найдет его в строке «победа». Наоборот, выражение «\Bбеда» найдет совпадение в строке «победа» но проигнорирует слово «беда».

Если необходимо повторить через «*» или «+» последовательность длиннее одного символа, то необходимо использовать круглые скобки для указания ее границ. Например, «((^\s+)[A-Я][a-я]+)» совпадает с любым количеством русских «слов», написанных с большой буквы и разделенных любым числом пробелов (начинаться она может с начала строки или с любого пробела в ней). К знакам повторения относится также «?», означающий ноль или один вариант. Любое количество повторений можно записать, используя фигурные скобки, например «{3,5}» – от 3 до 5 раз, «{3,}» – не менее трех раз, «{3}» – ровно три раза.

Другой часто используемый случай – возможность записи альтернативных вариантов, которая делается через вертикальную черту. Например, выражение «резвый|быстрый|оживленный» совпадает с одним из этих трех синонимов. Одним из вариантов может быть пустая альтернатива: «(способный\s+)студент» совпадает со строками «способный студент» и «студент».

Использование регулярных выражений позволяет резко сократить запись всех вариантов возможных ответов в случае, если в ответе

встречается более одного альтернативного участка, так как количество правильных ответов, которые надо вводить строками, растет мультипликативно. При наличии более двух альтернативных участков создание вопроса перечислением всех правильных ответов без регулярных выражений становится слишком трудоемким.

В вопросах по языкам программирования можно задавать произвольные имена переменных. Например, регулярное выражение «int\s+[a-zA-Z_]\w*» означает объявление целочисленной переменной с любым допустимым именем на языке Си. Например, «int counter».

Отдельные части регулярного выражения часто берут в круглые скобки. Части в круглых скобках называются подвыражениями. Например, в «(ab|cd)ef(gh)» есть два подвыражения. Они нумеруются в порядке открывания скобок, начиная с 1. Всё выражение в целом имеет номер 0. К подвыражениям в дальнейших частях регулярного выражения можно «обращаться» по их номерам с помощью обратных ссылок (например, «\1» или «\2»). Обратная ссылка будет совпадать с тем же набором символов, с которым совпало подвыражение с указанным номером (в отличие от повторения альтернативы, которое повторяет именно альтернативу, а не однажды выбранный вариант).

Рассмотрим пример из программирования. Студенту предлагается написать конструкцию на языке Си для увеличения значения уже объявленной переменной на любое целое положительное число, используя операцию сложения. Имя переменной может быть любым. Шаблон ответа будет выглядеть так: «int\s+([a-zA-Z_]\w*)\s*=\s*\1\s*\+\s*\d+\s*»;. Такой шаблон будет совпадать со строками «counter = counter + 1» или «x = x + 123»;. Студент имеет полную свободу в выборе имени переменной и числа. Для того чтобы в правой части равенства студент написал то же имя переменной, что и в левой части, используется обратная ссылка «\1». Она работает следующим образом. Имя переменной слева является

первым подвыражением. Обратная ссылка на него совпадает только со строкой, совпавшей с самим подвыражением. Обратите внимание, что это не то же самое, что повтор подвыражения – при повторе совпадения не обязаны повторяться. Другими словами, «([a-z]){2}» может совпасть с любой комбинацией двух букв, в то время как «([a-z])\1» совпадает лишь с парой одинаковых букв.

Если вы группируете несколько символов (для повторения или альтернативы) с помощью круглых скобок, однако на это подвыражение нет обратных ссылок, можно заменить круглые скобки «(...)» на конструкцию «(?: ...)». Дело в том, что для запоминания информации о подвыражениях требуется дополнительная память и время при поиске совпадения. Регулярные выражения с такими оптимизациями обычно работают быстрее. Кроме того, если вы используете обратные ссылки в выражении с большим количеством скобок, то активное использование группировки через «(?: ...)» позволяет вам иметь меньшие номера подвыражений в обратных ссылках (снижая шанс ошибиться и уменьшая вероятность их изменений при редактировании выражения, если появляются (или удаляются) новые пары скобок для группировки).

Часть символов (не букв и цифр, а различных знаков), как видно из предыдущих абзацев, в регулярных выражениях считается служебной. Если необходимо, чтобы такой символ (точка, звездочка и т.д.) попал в ответ, перед ним необходимо поставить знак обратной дробной черты – например, «\+» обозначает не повторение, а просто символ сложения. Это касается и самого знака обратной дробной черты – если он нужен вам в ответе, его необходимо продублировать. Вне квадратных скобок обратную дробную черту необходимо ставить перед следующими символами: «[^]\$.[]()*+{», внутри квадратных скобок – перед «[^]-[]». Чтобы не запоминать этот перечень, можно руководствоваться следующим правилом: можно безопасно ставить обратную дробную черту перед любым символом

кроме букв и цифр, если этот символ вам нужен буквально – обратная дробная черта перед любым значком снимает его специальное значение. И наоборот, никогда не ставьте обратную дробную черту перед буквой или цифрой, если вы не знаете, зачем вы это делаете: буквы и цифры получают специальное значение, если перед ними стоит обратная дробная черта.

Более подробно о синтаксисе регулярных выражений можно прочитать в книгах Фридла [4] (с объяснением всех тонкостей) и Форта [5] (более простым языком).

По данным английских исследователей, регулярные выражения удобно и эффективно применяются к ответам, являющимся по смыслу короткими предложениями на естественном языке, давая результаты в среднем не хуже чем при проверке учителем [6]. В этом исследовании также делается вывод, что регулярные выражения работают не хуже чем специализированные программы работы с шаблонами для естественных языков (см. ниже вопрос PMatch), при этом регулярные выражения легче изучаются, по ним больше литературы и специалистов.

2. Использование тестовых вопросов с подсказками в обучении

Пионером использования автоматически генерируемых подсказок в тестовых вопросах с регулярными выражениями стал Джозеф Резо (Университет Ренна-2), создавший специальный тип вопросов RegExp [7]. Сценарий использования подсказок такой: обучаемому показывается правильное начало ответа (если оно есть) и место, с которого его ответ перестал совпадать с шаблоном. Если он не знает, как продолжить ответ с этого места, он может запросить подсказку (при этом оценка за ответ может снижаться). Подсказкой может быть либо следующий корректный символ, либо следующая лексема (под лексемой понимается последовательность символов с определенным совокупным значением [8]). Для естественных языков примерами лексем

являются слова и знаки препинания (при этом троеточие, например, – одна лексема); для языков программирования лексемы это ключевые слова, идентификаторы, числовые константы, литералы, знаки операций и скобки. Подсказка следующей лексемы может также показать завершение лексемы, если ошибка сделана в ее середине.

Преимуществом таких автоматических подсказок по сравнению со встроенными во все вопросы Moodle подсказками, вводимыми автором вопроса, в том, что автоматически генерируемые подсказки учитывают место, где студент сделал первую ошибку, и всегда могут помочь ему продолжить работу над поиском правильного ответа. В то же время автоматические генерируемые подсказки не требуют дополнительной работы от автора вопроса (кроме указания языка ответа для правильного разбиения на лексемы).

Например, если студент не может корректно написать слово «цвета» на английском языке и правильное начало его ответа «colo», он может запросить подсказку до конца лексемы и получить продолжение «rs», так как это наиболее короткое завершение данного слова. Подсказка может показываться отдельной строкой с выделением цветом подсказки, а также правильной и ошибочной частей ответа (как реализовано в разработанном авторами вопросе PReg) или же добавляться непосредственно в ответ студента (как это реализовано в вопросе RegExp).

3. Аналогичные программы

PMatch – тип вопроса для СДО Moodle, разработанный The Open University. Он позволяет студентам писать короткие (до 20 слов) ответы, которые затем сравниваются с шаблонами ответов, записанными в специальном синтаксисе PMatch. Для задания вариантов ответов студента используются специальные английские ключевые слова. Возможно описание перестановок и пропусков частей ответов [9].

К недостаткам PMatch относится то, что этот вопрос не может

показывать правильные части частично правильных ответов и генерировать подсказки для завершения таких ответов. Используемый в нем язык шаблонов мало распространен и требует специального изучения. Более того, этот язык шаблонов вопроса PMatch ориентирован на вопросы с ответами на естественных языках (английский и близкие к нему по грамматической структуре языки) и не может использоваться для оценки ответов на языках программирования, а также на естественных языках, значительно отличающихся от английского.

Для системы Moodle также разработан тип вопроса, работающий с регулярными выражениями. Он называется RegExp [7], его автор – Joseph Rézeau, преподаватель английского языка из Франции.

В данном типе вопроса впервые были реализованы идеи нахождения частичного совпадения ответа с регулярным выражением и автоматической генерации подсказок в виде продолжения строки с того места, где студент сделал первую ошибку.

Однако большим недостатком типа вопроса RegExp является выбранный алгоритм поиска совпадения и генерации подсказки, основанный на попытке построить множество всех строк, совпадающих с шаблоном (регулярным выражением). Это ограничивает синтаксис используемых регулярных выражений (так, невозможно использовать повторение с «бесконечной» границей в виде квантификаторов «+» и «*»), а также значительно замедляет работу (вплоть до «зависания») для шаблонов с большим количеством строк, с которыми они могут совпадать. Конкретное количество обрабатываемых строк зависит от конфигурации сервера, в опыте авторов вопросы невозможно использовать при превышении границы в 1000 строк, что на практике достигается достаточно часто – всего лишь десять альтернативных участков ответа с двумя альтернативами в каждом (например, необязательные пробелы) дают 1024 варианта строки.

Интерфейс редактирования вопросов RegExp является неудоб-

ным: первым из всех ответов должна быть обычная строка, которая не интерпретируется как регулярное выражение и используется для показа студентам в качестве правильного ответа (очевидно, что показывать регулярное выражение с этой целью бессмысленно). При этом она участвует в генерации подсказок и иногда может быть показан не самый короткий вариант продолжения частично правильного ответа.

Система подсказок в RegExp также менее наглядна. Из ответа студента он удаляет неправильное окончание и ставит на его место подсказку, при этом не видно границы между правильной частью ответа и добавленной подсказкой. Это также слегка стимулирует механическое нажатие кнопки подсказки обучаемыми до получения правильного ответа.

К достоинствам RegExp следует отнести хорошую документацию для новичков в регулярных выражениях (на английском языке) и широкую распространенность вследствие более раннего появления.

4. Разработанный тип вопроса Preg

Открытые вопросы с описанием правильных ответов в виде шаблонов и автоматически генерируемыми подсказками были реализованы авторами в модуле типа вопроса Preg для СДО Moodle [10]. Этот модуль позволяет создавать и редактировать вопросы, оценивать ответы студентов и показывать комментарии к ошибкам.

На рис. 1 показаны основные настройки в форме редактирования

вопроса Preg для приведенного в начале примера о цветах.

Основные настройки содержат текст вопроса, общий отзыв, метод поиска совпадений («движок») с регулярными выражениями и форму записи (нотацию) регулярных выражений. Метод поиска совпадений «недетерминированные конечные автоматы» опирается на код собственной разработки, позволяет определять частичные совпадения и использовать подсказки. Реализация основывается на абстракции недетерминированных конечных автоматов с тегами, предложенной Вилли Лаурикари [11]. Метод удобно использовать для тренировочных тестов. К его недостаткам следует отнести неполную поддержку синтаксиса perl-совместимых регулярных выражений (к настоящему времени – самый богатый по возможностям диалект регулярных выражений [4]) – не поддерживаются сложные утверждения и некоторые другие редко используемые виды операций, а также возможное наличие ошибок в редких случаях. Движок «расширение preg для php» основан на существующей библиотеке perl-совместимых регулярных выражений PCRE [4]. Он не позволяет пользоваться подсказками, но полностью поддерживает perl-совместимые регулярные выражения и более надежен, будучи активно используемым в течение многих лет.

Нотации регулярного выражения дают автору вопроса определенную гибкость в форме записи правильного ответа. В настоящее время поддерживают две нотации perl-совместимых регулярных выражений: обычная и расширенная (для длинных выражений – позво-

ляет легко вставлять комментарии и форматировать код пробелами), а также нотация Moodle Shortanswer, позволяющая использовать ответы из обычных вопросов типа «Короткий ответ» СДО Moodle с возможностью генерации автоматических подсказок без какого-либо знания регулярных выражений.

На рис. 2 показаны дополнительные настройки вопроса. Они позволяют управлять доступностью подсказок следующего символа или лексемы (подсказки могут быть нежелательны в контрольных тестах), а также устанавливать снижение оценки (штраф) за их использование. Если вы хотите использовать подсказку следующей лексемы, то следует задать язык, на котором вводятся ответы на вопрос (для корректного разбиения его на лексемы). На данный момент поддерживается английский язык, языки программирования Си и Си++, а также язык форматной строки функции printf, используемой в языке Си. Добавление нового языка достаточно просто для квалифицированного программиста.

Традиционно, регулярные выражения использовались для поиска совпадения внутри текста, т.е. успешным считалось совпадение всего регулярного выражения (задается преподавателем) с любой частью текста (ответ обучаемого). Вопрос сохраняет это поведение по умолчанию, позволяя учителю описать обязательную часть ответа, а не весь ответ. Однако во многих случаях преподаватель хочет задать регулярное выражение для всего ответа: в этом случае следует включить настройку «точное совпадение». Если точное совпадение отключено, оно может быть достигнуто в некоторых ответах преподавателя с использованием простых утверждений «^» и «\$», описанных выше в разделе про регулярные выражения. Такой подход позволяет дать преподавателю гибкость, если он ее хочет, в то же время упрощая работу начинающему составителю вопросов возможностью использовать настройку точного совпадения для всех ответов.

Автор вопроса также может ввести правильный ответ для показа

Рис. 1. Пример формы редактирования для вопроса с ответом на английском языке – основные настройки

Рис. 2. Пример формы редактирования для вопроса с ответом на английском языке – дополнительные настройки

Рис. 3. Пример задания шаблона правильного ответа

Рис. 4. Пример задания шаблона типичного неправильного ответа и отзыва с подстановкой ошибки из шаблона

обучаемому после теста (в виде текста, чтобы не показывать обучаемому регулярное выражение), но это не обязательно при использовании движка на недетерминированных конечных автоматах – система генерации подсказок может обеспечить генерацию правильного ответа самостоятельно, если автор оставит вопрос пустым. Настраивается также чувствительность ответа к регистру (прописные и строчные буквы).

Форма задания ответа преподавателя показана на рис. 3. В данном случае шаблон немного усложнен конструкциями «\s+» для подде-

ржки любого количество подряд идущих пробелов между словами. Автор вопроса вводит регулярное выражение в поле «вариант ответа», указывает оценку (возможен ввод частично правильных ответов) и отзыв для конкретного ответа (есть также общий отзыв на вопрос, показываемый по окончании тестирования).

Ответы с оценкой менее 100% (вплоть до нулевой оценки) обычно используются с отзывами в случае, если преподаватель хочет показать специальное сообщение для определенных ошибок. Например,

шаблон ответа, «отлавливающего» типичную ошибку нарушения требований к имени переменной на языке Си, показан на рис. 4.

Обратите внимание, что это выражение совпадает не только с ошибочными, но и с правильными именами переменных (\S+ это любая последовательность непробельных символов). Поскольку ответы преподавателя проверяются по порядку до первого точного совпадения, то первый вариант ответа будет совпадать для правильных написаний, а до второго дойдет только в ошибочном случае. Если же точного совпадения нет, то для определения подсказки алгоритм ищет такое частичное совпадение с правильным ответом (по умолчанию – только на 100%-ю оценку), которое ближе всего к завершению (нужно добавить наименьшее количество символов для точного совпадения). Вы можете контролировать границу между «правильными» (к которым ведут подсказки) и неправильными ответами при помощи специальной настройки, но обычно она достаточно высока – едва ли вы захотите, чтобы следование подсказкам привело обучаемого к ответу с оценкой 20%.

Обратите также внимание на управляющую последовательность «\${S1}» в комментарии. Она будет заменена на совпадение с первым подвыражением (см. выше раздел об использовании регулярных выражений). В отзыв подставляется часть ответа, совпавшая с ним. Например, студент может увидеть вот такой отзыв (рис. 5).

Пример получения автоматически сгенерированной подсказки следующей правильной лексемы показан на рис. 6: правильная часть ответа выделена зеленым, неправильная – красным, а подсказка – желтым цветом фона.

5. Результаты использования вопросов Preg в преподавании дисциплины «Основы программирования» в ВолГТУ

Разработанный тип вопроса Preg внедрен в учебный процесс на дисциплине «Основы программи-

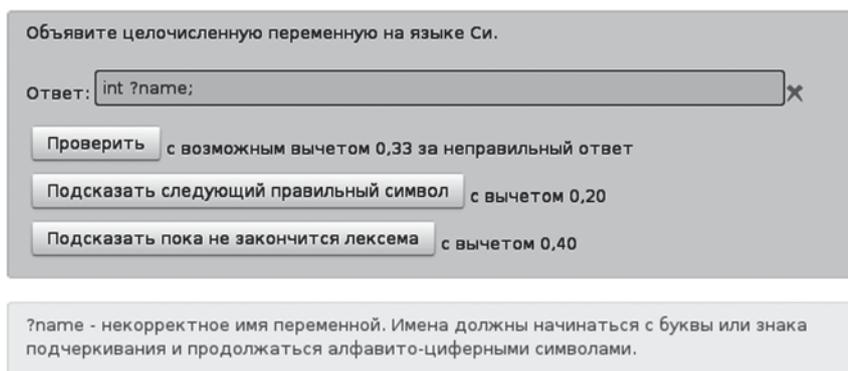


Рис. 5. Пример получения студентом отзыва на ответ, содержащий предусмотренную преподавателем ошибку

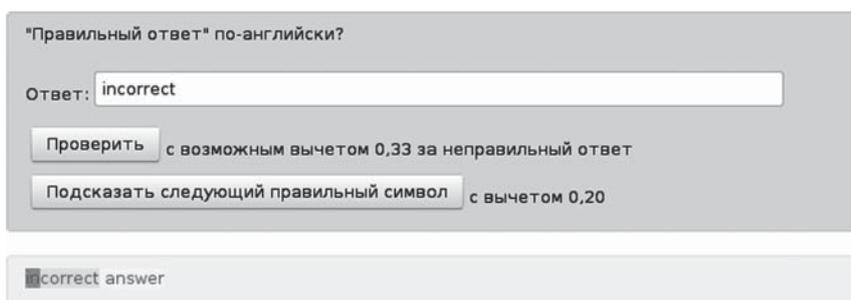


Рис. 6. Пример получения студентом подсказки следующей правильной лексемы

рования» на 1-м курсе факультета электроники и вычислительной техники Волгоградского государственного технического университета (ВолгГТУ). В рамках разработанного набора тестов вопросы типа Preg играли несколько ролей. Часть вопросов была разработана для тренировок с подсказками. Вопросы типа Preg также использовались в контрольных тестах, а также тренировочных тестах без подсказок (симулирующих проведение контрольных) для проверки ответов студента по шаблону. Эти вопросы для студентов были почти неотличимы от стандартных вопросов с открытым ответом, за исключением того факта, что в конце тестирования показывали, до какого конкретно места ответ был правильным.

В общей сложности для первого семестра было разработано 253 контрольных вопроса, 98 тренировочных вопросов с подсказками и 49 тренировочных вопросов без подсказок. Тренировочные вопросы покрывали 17 тем по 6 лабораторным работам. Для второго семестра было разработано 227 контрольных вопросов, 82 трени-

ровочных вопроса с подсказками и 61 тренировочный вопрос без подсказок. Во втором семестре тренировочные вопросы покрывали 8 тем по 4 лабораторным работам.

В конце семестра было проведено анкетирование 77 студентов, которые пользовались тренировочными тестами. Следующие возможности вопросов типа Preg были отмечены как наиболее полезные:

- 1) текстовые комментарии к типичным ошибкам с подстановкой самих ошибок;
- 2) подсказка следующей лексемы;
- 3) выделение правильной и неправильной частей ответа цветом.

95% студентов выразили желание иметь тренировочные тесты Preg в следующем семестре. Они отмечали, что подсказки полезны в вопросах на перевод чисел в строку и наоборот, действия со строками, вызовы функций с передачей в них массива и т.д. Примерно такое же количество студентов выразило желание использовать тип вопроса CorrectWriting [12].

Были проанализированы данные о попытках прохождения тренировочных тестов с подсказками, сохраняющиеся в базе данных системы Moodle. В течение семестра тренировочными тестами Preg пользовались 73 студента. В общей сложности было дано 804 ответа на вопросы. Подсказками пользовались 57 студентов (78%), остальные предпочитали сразу завершать попытку ответа на вопрос и смотреть комментарий.

В табл. показано использование студентами возможностей вопроса Preg в разрезе их успеваемости по предмету (в рамках 100-балльной рейтинговой системы оценивания). Отдельно выделены студенты, получившие минимальный балл положительной оценки (61). Из табл. видно, что пользовались подсказками активнее всего студенты, обучающиеся на «отлично» и «удовлетворительно». Если низкий процент использования подсказок среди тех, кто получил минимальную удовлетворительную (61 балл), а также неудовлетворительную оценку объясняется тем, что эти студенты вообще занимались менее активно, то ситуация с «хорошистами» более сложная. Они достаточно активно использовали сам вопрос, но не автоматически сгенерированные

Таблица использования тестов Preg студентами в зависимости от полученной итоговой оценки

	Отлично 90..100 баллов	Хорошо 75..89 баллов	Удовлетворительно 62...74 балла	Минимальный балл – 61	Неуд.	Всего
Количество	23	25	7	33	11	99
Пробовали preg	78%	80%	71%	70%	64%	74%
Отвечали на preg	78%	72%	71%	58%	64%	68%
Использовали подсказку символа	74%	64%	71%	48%	27%	58%
Использовали подсказку лексемы	74%	56%	71%	52%	36%	58%

подсказки. Возможно, это связано с прилежным, но формальным подходом к учебе, что могло в итоге привести к получению оценки «хорошо» теми, кто мог бы получить и «отлично».

С точки зрения преподавателя, использование типа вопроса Preg имеет несколько преимуществ:

1) значительная экономия трудозатрат на создание вопросов при наличии в ответе нескольких мест с альтернативными вариантами написания, за счет резкого уменьшения количества вводимых ответов;

2) устранение значительного числа ложноотрицательных оценок (например, за счет учета лишних

или необязательных пробелов);

3) более активное использование возможностей самостоятельной подготовки студентов, за счет использования автоматически генерируемых подсказок следующего символа и следующей лексемы, которые позволяют студенту самостоятельно найти правильный ответ при любом количестве ошибок.

Результаты опроса студентов ВолгГТУ, а также отзывы преподавателей показывают, что использование открытых вопросов с подсказками позволяет улучшить качество самостоятельной подготовки студентов при изучении языков программирования или естествен-

ных языков. Снижаются затраты времени преподавателей на составление вопросов, проверку случаев ложноотрицательных ответов и подсказки студентам, которым материал дается с трудом. Автоматические подсказки в реализованном модуле Preg могут быть использованы даже преподавателями, не знакомыми с регулярными выражениями при использовании нотации «Moodle Shortanswer».

Разработанный модуль распространяется на условиях лицензии GPL через страницу сайта СДО Moodle https://moodle.org/plugins/pluginversions.php?plugin=qtype_preg.

Литература

1. Статистика Moodle [Electronic resource]. – URL: <https://moodle.org/stats/> (дата обращения: 26.10.2014).
2. Фандей В.А. Использование модульной объектно-ориентированной динамической учебной среды Moodle в преподавании межкультурной коммуникации // Вестник Московского университета. Серия 19: Лингвистика и межкультурная коммуникация. – 2011. – № 4. – С. 125–130.
3. Соловов А.В. Электронное обучение: проблематика, дидактика, технология. – Самара: Новая техника, 2006. – 464 с.
4. Фридл Дж. Регулярные выражения / пер. с англ. – 3-е изд. – СПб.: Символ Плюс, 2008. – 608 с.
5. Форта Б. Регулярные выражения. – М.: ИД «Вильямс», 2005. – 184 с.
6. Butcher P.G., Jordan S.E. A comparison of human and computer marking of short free-text student responses // Computers & Education. – 2010. – № 55. – P. 489–499.
7. Regular Expression Short-Answer question type [Electronic resource]. – URL: http://docs.moodle.org/27/en/Regular_Expression_Short-Answer_question_type (date access: 26.10.2014).
8. Ахо А., Сети Р., Ульман Дж. Компиляторы. Принципы, технологии, инструменты. – М.: ИД «Вильямс», 2003. – 769 с.
9. Pattern-match question type detailed documentation [Electronic resource]. – URL: https://docs.moodle.org/27/en/Pattern-match_question_type_detailed_documentation (date access: 26.10.2014).
10. Сычев О.А. Использование регулярных выражений для проверки ответов на тестовые вопросы, предоставляемых в виде фрагмента кода программы / О.А. Сычев, Д.В. Литовкин // Информационные технологии в образовании, технике и медицине: матер. междунар. конф., 21–24 сент. 2009 / ВолгГТУ [и др.]. – Волгоград, 2009. – С. 45.
11. Laurikari V. NFAs with tagged transitions, their conversion to deterministic automata and application to regular expressions // Proceedings of the 7th International Symposium on String Processing and Information Retrieval. – IEEE, 2000. – P. 181–187.
12. Сычев О.А. Автоматическое определение ошибок в порядке расположения лексем в ответах на вопросы с открытым ответом в СДО Moodle / О.А. Сычев, Д.П. Мамонтов // Открытое образование. – 2014. – № 2. – С. 79–88.