УДК 004.91 **Информационные технологии**

Великоредчанина Ирина Сергеевна, *студент, Хакасский государственный университет им. Николая Федоровича Катанова, Россия, г. Абакан*

Голубничий Артем Александрович, *научный руководитель, старший преподаватель кафедры ПОВТиАС, Хакасский государственный университет им. Николая Федоровича Катанова, Россия, г. Абакан*

СОЗДАНИЕ СИСТЕМЫ ТЕСТИРОВАНИЯ ЗНАНИЙ ПО ДИСЦИПЛИНЕ «ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ» СРЕДСТВАМИ R/EXAMS

Аннотация: В статье анализируется система составления экзаменов на языке R посредством пакета exams. Рассматриваются особенности генерации экзаменов, анализируется возможность создания материалов для проверки знаний по дисциплине парадигмы программирования. Описывается процесс интеграции экзаменов в систему управления обучением Moodle.

Ключевые слова: генерация экзаменов, язык R, Moodle, электронное обучение, пакет exams.

Abstract: The article analyzes the system for writing exams in the R language using the exams package. The features of the generation of exams are considered, the possibility of creating materials for testing knowledge in the discipline of the programming paradigm is analyzed. The process of integrating exams into the Moodle Learning Management System is described.

Keywords: generation of exams, R language, Moodle, e-learning, exams package.

Во время организации учебного процесса часто возникает задача генерации материалов, используемых для проверки знаний. В зависимости от формы проведения экзамена, возникает необходимость создания проверочных

работ в электронном или бумажном формате. Для исключения возможности списывания ответов у коллег по обучению применяют различные технологии, наиболее частыми вариантами которых могут служить:

- случайный выбор заданий из некоторый базы вопросов;
- случайное перемешивание вариантов ответов к вопросу (при этом можно использовать большое количество разных неправильных ответов, а представлять в качестве возможных вариантов лишь несколько (4-5) вариантов);
- случайный выбор параметров (чисел, текстовых блоков, графиков и т.д.) для формирования уникальных вопросов.

Все технологии создания уникального контента для проверки знаний можно реализовать посредством пакета exams [1] языка программирования R [2]. Весь процесс создания экзаменов (материалов для проверки знаний) можно представить в виде набора шагов:

1. Создание базы вопросов.

2. Кодирование вопросов с целью приведения к формату, понимаемому системой (.Rmd или .Rnw).

3. Генерация вопросов в нужный формат экзамена.

Рассмотрим каждый шаг для создания системы проверки знаний по дисциплине «Парадигмы программирования».

Создание базы вопросов. Этап начинается с анализа материала, планируемого к проверке. С учетом того, что дисциплина парадигмы программирования предполагает ознакомление студентов с теоретическим материалом, показывающим разнообразие и отличие некоторых особо популярных парадигм программирования, то и вопросы, планируемые к проверке, должны охватывать данные парадигмы. Все вопросы для проверки разделены на 9 тем: Общие сведения о парадигмах программирования; Императивное программирование; Структурное программирование; Объектно-ориентированное программирование; Векторное программирование; Функциональное программирование; Параллельное программирование; Грамотное программирование; Метапрограммирование.

Для каждой темы планируется не менее 10 вопросов, при этом для проверки знаний целесообразно использовать не все вопросы, а лишь некоторое их количество, что в свою очередь даст возможность проверить знания по данной теме, одновременно с этим оставив возможность для итогового контроля знаний по дисциплине.

Кодирование вопросов. На данном этапе содержание вопроса переводится в формат Rmd или Rnw, принципиально данные форматы отличаются только синтаксисом, в первом случае используется язык упрощенной разметки Markdown со вставками кода на языке R, во втором случае вместе Markdown используется LaTeX.

Вопросы являются основной структурной частью экзамена, именно из вопросов составляется уникальный вариант экзамена, при этом структура вопроса может содержать до четырех структурных элементов: генерация данных; текст вопроса; решение; метаинформация. При этом обязательными являются текст вопроса и метаинформация. На рисунке 1 представлен один из вопросов закодированный в формате Rmd содержащий три из четырех элементов вопроса. Как видно данный вопрос не включает генерацию данных.

```
Question
========
Укажите верные высказывания

Answerlist
----------
* Новая парадигма программирования отменяет предыдущую
* Существует единый подход к классификации языков по парадигмам программирования
* Парадигмы программирования могут сочетаться
* Язык программирования может поддерживать не более 3 парадигм
* Язык программирования может поддерживать неограниченное множество парадигм

Solution
========

Answerlist
----------
* Не верно. В отличии от парадигм в науке, парадигмы программирования не отменяют друг друга
* Не верно. Существуют разные подходы к классификации и подклассификации парадигм и, соответственно, языков программирования по парадигмам
* Верно. Парадигмы могут дополнять друг друга. Так элементы структурного программирования встречаются в ООП и др. парадигмах
* Не верно. Такие языки как R, Java, Python, C++ и многие другие могут поддерживать большинство известных парадигм программирования
* Верно. Такие языки как R, Java, Python, C++ и многие другие могут поддерживать большинство известных парадигм программирования

Meta-information
================
exname: Общие сведения 01
extype: mchoice
exsolution: 00101
exshuffle: 5
```

Рисунок 1 – Листинг вопроса

Рассмотрим несколько подробнее структурные элементы вопроса. Первой частью вопроса (не обязательной) является генерация данных. Генерация данных особо полезна, когда есть необходимость создания вопроса, построенного на случайных данных. Данный подход наиболее актуален, когда речь идет о вычислительных задачах. В процессе генерации данных помимо определения изначальных переменных также можно генерировать и ответ. В этом случае можно полностью автоматизировать процесс оценки.

Вторая часть вопроса – текст вопроса является обязательной составляющей. На рисунке 1 представлен вариант, когда текст вопроса предполагает статичное содержание и варианты ответа (Answerlist), такая структура типична для вопросов типа выбора одного из множества значений или множественного выбора.

Третья часть вопроса – решение, также, как и генерация данных относится к необязательным элементам. Решение является важным элементом, показывающим порядок действий необходимый для вычисления итогового ответа, или, в случае с текстовым ответом, может содержать обоснование правильности (неправильности) того или иного выбора.

Метаинформация относится к последнему структурному элементу вопроса и является обязательной частью. Метаинформация отличается для разных типов вопросов, пример метаинформации для вопроса представлен на рисунке 2.

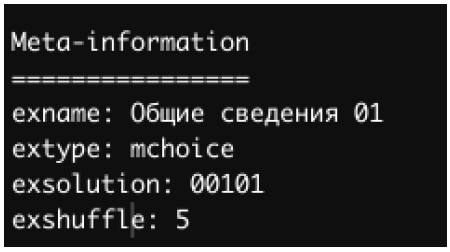

Рисунок 2 – Структурный элемент метаинформация

В метаинформации содержится наименование вопроса, его тип, правильные ответа из списка возможных вариантов и количество вариантов, предлагаемых для пользователя.

Генерация вопросов в нужный формат. На последнем этапе необходимо определиться с форматом, используемым для проведения оценки знаний. Особенность системы exams заключается в том, что можно генерировать различные типы экзаменов. Генерация экзамена происходит с помощью команд типа exams2___(), где вместо подчеркивания вписывается необходимый формат. На данный момент поддерживаются следующие варианты генерации данных: arsnova, blackboard, canvas, html, lops, moodle, nops, openolat, pandoc, pdf, qti12, qti21, tcexam.

Как видно, значительная часть вариантов поддерживает различные электронные системы обучения, одной из самых популярных свободных систем управления обучения является Moodle. Возможность конвертирования вопросов в Moodle в значительной степени упрощает процесс проведения контроля знаний. Функция exams2moodle() принимая на вход перечень вопросов и некоторые дополнительные аргументы генерирует нужное количество вопросов и сохраняет их в XML файл легко читаемый экспортом вопросов Moodle. Листинг исходного кода, используемый для генерации экзамена по первому разделу дисциплины парадигмы программирования представлен на рисунке 3.

```
paradigm_exam_01 <- c("01-01.Rmd", "01-02.Rmd", "01-03.Rmd", "01-04.Rmd",
                      "01-05.Rmd", "01-06.Rmd", "01-07.Rmd", "01-08.Rmd",
                      "01-09.Rmd", "01-10.Rmd")

exams2moodle(sample(paradigm_exam_01, 5), n = 10, name = "R-exams-1")
```

Рисунок 3 – Генерация вариантов экзамена

Как видно из программного кода из изначального набора данных по вопросам случайным образом выбирается пять вопросов и генерируется 10 вариантов с разным порядком ответов. На выходе получается исходный файл R-exams.xml, используемый для создания теста по первой теме. Аналогичным образом обрабатываются все остальные вопросы.

В ходе данной работы была рассмотрена система создания экзаменов с использованием языка программирования R и пакета exams. Были составлены вопросы для проверки знаний по дисциплине парадигмы программирования.

Библиографический список:

1. CRAN – Package exams [Электронный ресурс] URL: https://CRAN.R-project.org/package=exams (дата обращения: 10.08.2020).

2. The Comprehensive R Archive Network [Электронный ресурс] URL: https://cran.r-project.org (дата обращения: 10.08.2020).