

Система управления оценочными средствами

К. В. Баталин, Г. Э. Яхьяева

*Новосибирский государственный университет
Новосибирск, Россия*

Аннотация

В статье описана программная система управления оценочными средствами преподавателя, позволяющая в полуавтоматическом режиме создавать комплект оценочных документов.

В основе алгоритма генерации оценочного документа лежит алгоритм кластеризации категориальных данных. Представлена модификация алгоритма CLOPE, позволяющая автоматически определять необходимое число кластеров в зависимости от входных данных. Также данная модификация решает проблему маленьких кластеров и проблему категориальной кластеризации числовых атрибутов.

Описан итерационный алгоритм формирования комплекта оценочных документов, состоящий из заданного пользователем числа однотипных по структуре, но уникальных по содержанию вариантов оценочных документов.

Ключевые слова

оценочный документ, комплект оценочных документов, кластеризация, категориальные данные, CLOPE

Для цитирования

Баталин К. В., Яхьяева Г. Э. Система управления оценочными средствами // Вестник НГУ. Серия: Информационные технологии. 2020. Т. 18, № 2. С. 5–14. DOI 10.25205/1818-7900-2020-18-2-5-14

Assessment Means Management Software

K. V. Batalin, G. E. Yakhyeva

*Novosibirsk State University
Novosibirsk, Russian Federation*

Annotation

The program system of the management of assessment tools for educational process described in the article. The system reduces the time that the teacher spends on the preparation of test materials, allowing you to create a set of assessment documents in a semi-automatic mode.

The system implements the ability to create and manage task banks. Each task in the bank has a set of attributes that are involved in the process of generating assessment documents. The system also implements the ability to automatically generate the required number of variants of the assessment documents. At the same time, the algorithm for generating a set of assessment documents works in such a way that, on the one hand, one set includes the most similar variants of assessment documents in structure, and on the other hand, each assessment document in the set is unique.

The algorithm for generating a set of assessment documents is based on the clustering algorithm for categorical data. In this paper a modification of the CLOPE algorithm was submitted, which allows you to automatically determine the required number of clusters, depending on the input data. Also, this modification solves the problem of small clusters and the problem of categorical clustering of numerical attributes.

The paper also describes an iterative algorithm for the generation of a set of assessment documents.

Keywords

assessment document, a set of assessment documents, clustering, categorical data, CLOPE

For citation

Batalin K. V., Yakhyeva G. E. Assessment Means Management Software. *Vestnik NSU. Series: Information Technologies*, 2020, vol. 18, no. 2, p. 5–14. (in Russ.) DOI 10.25205/1818-7900-2020-18-2-5-14

Введение

Современное общество с каждым днем овладевает все большим набором знаний. Наука и технологии, а следовательно, и образование развиваются. Образовательные курсы меняют направленность, добавляются новые темы, покрываются новые компетенции, старые подходы адаптируются под современные методы. Все это влечет за собой и изменение способов оценивания и тестирования знаний студентов [1].

Программа учебных курсов регулярно модифицируется, поэтому преподавателям приходится каждый раз не только создавать новые оценочные материалы, но и подготавливать их для проверочных работ: группировать задания по множеству критериев, составлять похожие между собой варианты проверочных работ. Еще одной причиной регулярного составления оценочных материалов является общедоступность старых вариантов проверочных работ. Задания и вопросы публикуются студентами прошлых курсов, что снижает эффективность повторного использования большинства из них.

Составление проверочных материалов – трудоемкое занятие, которое отнимает значительное время у преподавателей, поэтому актуальным становится создание автоматизированных систем, облегчающих работу преподавателя с оценочными средствами.

Обзор существующих инструментальных средств

Все существующие на сегодняшний день системы работы с оценочными материалами по степени автоматизации процесса подготовки оценочных материалов можно разделить на три класса [2]:

- 1) системы, решающие только проблемы оформления;
- 2) системы, решающие проблемы генерации банка вопросов;
- 3) системы, решающие проблемы генерации оценочных материалов на основе банка вопросов.

Рассмотрим каждую группу более подробно.

Системы, решающие проблемы оформления. Системы данного типа позволяют выбирать тип вопроса, вводить сам вопрос и ответ к нему, вводить дополнительную информацию, необходимую для оформления, генерировать файл с оценочными материалами, готовый к печати. В системах данного типа не идет речи об автоматическом разбиении вопросов, не говоря уже о генерации разных вариантов. Основная цель программных систем данного типа – оформление оценочных средств. Примерами систем данного класса являются следующие.

Testmoz¹ – англоязычный онлайн-сервис для создания оценочных материалов. Данный сервис предоставляет наибольшее число возможностей среди программных решений в данной категории. В системе есть поддержка разных типов вопросов, есть возможность проводить тесты онлайн, удобная форма обратной связи со студентами, подготовка документа к печати. Форма обратной связи позволяет преподавателю сообщить студенту об ошибках, дать советы по исправлению или уточнить ответы. В системе есть кастомизация формата печати, можно менять оформление.

EasyTestMaker² – онлайн генератор тестов. Есть возможность создавать различные тестовые задания: множественный выбор, заполнение пустых полей, короткий ответ и т. д. Есть функционал перевода оценочного документа в формат .pdf или .doc для дальнейшей распечатки.

Moodle (Modular Object-Oriented Dynamic Learning Environment)³ – одна из самых популярных систем электронного обучения [3]. Она переведена более чем на 100 языков, и ею

¹ testmoz.com

² www.easytestmaker.com

³ www.moodle.org

пользуются крупные университеты во всем мире. Модуль для проведения тестов (quizzes) в Moodle позволяет создавать тесты как контрольные работы (т. е. определять такие атрибуты, как дата начала, окончания, лимит времени и др.), разрабатывать тестовые задания и вносить их в банк вопросов, наполнять тест заданиями из банка вопросов и определять его внутреннюю структуру.

Системы, решающие проблемы генерации банка вопросов. Данный тип систем на вход чаще всего получает текст, по которому проводится тест. На выходе выдает список вопросов для этого текста.

Ярким представителем этой группы программных систем является система Quillionz⁴. Данная система старается автоматизировать весь процесс генерации оценочных материалов начиная составлением банков вопросов, заканчивая подготовкой проверочного материала к печати. Работа Quillionz основана на использовании нейронных сетей. Для работы с Quillionz необходимо загрузить в нее текст, в котором система с помощью искусственного интеллекта найдет ключевые слова, а далее сгенерирует вопросы к ним. Преподаватель может влиять на выбор ключевых слов: убирать предложенные системой ключевые слова, добавлять собственные.

Следующим шагом система создаст банк вопросов к выбранным ключевым словам. Вопросы могут быть по типу “дайте определение”, “выберите верный вариант ответа”, “объясните, почему верно суждение”. Пользователь системы выбирает понравившиеся вопросы, подготавливает документ с выбранными вопросами к печати. Повлиять на формат вопросов, на процесс генерации, на оформление итоговых документов преподаватель не может.

Система работает только с английскими текстами, вопросы получаются однотипными. В системе нет возможности генерировать несколько вариантов оценочных материалов по одному тексту.

Системы, решающие проблемы генерации оценочных материалов на основе банка вопросов. Данный тип систем на вход принимает список вопросов, на выходе выдает разбиение банка вопросов по нескольким вариантам

Генератор ТСПУ⁵ создан специально для Томского государственного педагогического университета. Генератор предоставляет онлайн-форму. В форме есть набор заранее зафиксированных полей. Есть вариант оформления билетов для 2-х или 3-х вопросов. Нет возможности указать собственное количество вопросов. Банк вопросов может быть задан как один для всех вопросов, так и отдельный для каждого вопроса. В качестве вопроса выступает текстовая строка. Каждый вопрос располагается на отдельной строке. На выходе система дает готовый для печати документ. Данная система поддерживает только один тип вопросов, строго заданы шаблон и оформление билетов, нет возможности задать собственное количество вопросов в билете, для выбора вопросов используется примитивный алгоритм выбора случайного.

Генератор билетов v2.0 (Ticket Generator)⁶ дает возможность не только генерировать проверочные материалы, но и управлять оценочными материалами. Преподаватель может подготавливать и сохранять банки вопросов, просматривать и редактировать их позднее, экспортировать и делиться ими с другими преподавателями. Есть возможность выбора разных шаблонов оформления для будущего проверочного материала: можно выбрать среди заранее подготовленных стандартных шаблонов или создать свой собственный. Есть возможность управления результатами генерации: изменение порядка вопросов, форматирование вопросов, задание разного типа вопросов, выбор категории вопроса. Помимо этого в системе для каждого вопроса можно установить сложность вопроса, объем ответа, оценку за правильный ответ. Но эти данные не будут участвовать в процессе генерации проверочного материала.

⁴ www.quillionz.com

⁵ <http://test.tspu.ru/gentickets>

⁶ <http://kaiu.narod.ru/Genbilet/Genbilet.html>

Они хранятся для того, чтобы преподаватель в будущем мог их просматривать и редактировать. Сам процесс генерации примитивный, преподаватель фиксирует категории для каждого вопроса, а далее случайным образом выбираются вопросы для теста.

ExamView⁷ – программный комплекс, позволяющий управлять оценочными материалами и генерировать готовые к печати вопросы, имеет большую гибкость в управлении материалами, поддержаны дополнительные функции работы с банками вопросов: слияние, автовыборка на основе критериев. Система имеет удобный пошаговый помощник генерации билетов. На каждом шаге преподавателю объясняется, какие данные необходимо ввести, как дальше эти данные будут использоваться. Помимо управления банками, система позволяет гибко манипулировать оформлением оценочных материалов. Предусмотренных шаблонов больше, чем в предыдущем варианте, интерфейс управления форматированием лаконичнее и понятнее.

Система Card_gen⁸ представлена в виде макроса Microsoft Word. У данного подхода есть свои преимущества и недостатки. Среда Microsoft Word знакома большинству современных пользователей, поэтому порог вхождения в систему Card_gen низкий, не требуются дополнительные инструкции по работе с макросом. Система не требует дополнительной установки, если у пользователя уже есть предустановленный Microsoft Word. Иначе использование системы может стать проблемой, так как среда Microsoft Word является платным решением. Списки вопросов система берет из другого документа, который находит в той же директории. Поддерживается только один формат документов с вопросами. По умолчанию алгоритм генерации основан на выборе случайного вопроса. Если необходимо поменять оформление оценочного материала или процесс генерации, то нужно делать исправление в исходном коде макроса. В макросе нет интерфейса для работы с генератором, что связано в первую очередь с концепцией макросов Microsoft Word.

Семантическая модель

Семантическая модель предметной области строится на основе четырехслойной модели представления знаний [4; 5]. Кортж K, σ, T^a, T^s, T^f является формальным представлением семантической модели предметной области, где K – множество прецедентов предметной области, σ – сигнатура, т. е. набор ключевых понятий, T^a – аналитическая теория предметной области, T^s – теория область и T^f – нечеткая теория области.

Разрабатываемая программная система относится к классу систем, решающих проблемы генерации оценочных документов на основе банка вопросов. Центральным понятием в разрабатываемой системе является понятие задания [6]. Каждое задание состоит из *тела задания* и набора *атрибутов*. Тело задания формализуется в виде предложения (или нескольких предложений) естественного языка, которые хранятся в системе в виде строковой величины и являются неделимым объектом.

Набор атрибутов у разных заданий может быть разным и формируется пользователем системы. Так, например, можно задать атрибут *тип*, который может принимать следующие значения: *вопрос*, *задача*, *тест*, *списочное задание*, *шаблонное задание* и т. п. Также очень полезным является атрибут *сложность*. Пользователь сам может устанавливать количество уровней сложности заданий, вводя допустимые значения данного атрибута. Аналитическая теория, т. е. набор аксиом, которым должно отвечать понятие задания, описана в работе [6].

С помощью описания соответствующих атрибутов задание можно привязать к учебному плану факультета: к отдельному модулю учебного плана, модулю дисциплин, одной отдельной дисциплине или к отдельной теме дисциплины. Также задание может быть привязано

⁷ <https://www.turningtechnologies.com/examview>

⁸ https://github.com/progsan/card_gen

к профессиональному стандарту или к отдельным его профессиональным функциям. Задание может быть привязано к некоторой компетенции или группе компетенций [7].

Банк заданий – это множество заданий и набор их атрибутов. Каждый банк заданий характеризуется своим набором атрибутов, т. е. в одном банке хранятся однотипные задания. У разных банков атрибуты могут отличаться.

Оценочный документ строится из конечного числа заданий. Число заданий в оценочном документе задается пользователем. В процедуре генерации оценочного документа могут участвовать несколько различных банков заданий. Алгоритм выбора заданий для оценочного документа в данной системе не является случайным, он учитывает атрибуты каждого задания.

На первом этапе формирования оценочных документов создается шаблон оценочного документа. Сначала задается количество заданий, входящих в формируемый шаблон. Далее пользователь настраивает шаблон каждого задания, которое он хочет получить в оценочном документе, т. е. задает банк, задания которого необходимо использовать для генерации, и маску задания.

Маска задания – это набор критериев, которым должны удовлетворять атрибуты задания в формируемом оценочном документе. Маска задается в виде строки, формат которой удовлетворяет грамматике

$$S \rightarrow S \vee T | T, T \rightarrow T \wedge M | M, M \rightarrow a | S,$$

где a – атом в формате

$$\text{name OPERATION value},$$

где *name* – название атрибута, *value* – значение для проверки атрибута, *OPERATION* – операция для проверки атрибута и значения.

$$\text{OPERATION} \in =, \neq, >, \geq, <, \leq.$$

В маске могут участвовать только те атрибуты, которые есть в выбранном банке заданий. Пример маски:

Тип = вопрос с открытым ответом, Сложность = легкое задание.

Набор масок задания и список выбранных банков заданий и задают шаблон оценочного документа.

Набор оценочных документов, соответствующих одному шаблону (т. е. вариантов оценочных документов), образует *комплект оценочных документов*. Число вариантов, которое необходимо получить на выходе работы системы, также задается пользователем. При этом алгоритм генерации комплекта оценочных документов работает так, что в один комплект, с одной стороны, входят максимально схожие по структуре варианты оценочных документов, а с другой стороны, каждый оценочный документ в комплекте уникален.

Алгоритм кластеризации категориальных данных

Для каждой маски задания создается *класс заданий*, удовлетворяющих данной маске, т. е. заданий, для которых все критерии маски выполнены. Далее, при формировании оценочного документа, из каждого класса заданий выбирается по одному заданию. Заметим, что одно и то же задание может подходить под разные маски, а значит, классы заданий могут пересекаться. Более того, в шаблоне оценочного документа для разных заданий могут быть заданы одинаковые маски. Следовательно, соответствующие этим маскам классы заданий будут совпадать. При формировании оценочного документа естественным является требование, чтобы оценочный документ не имел одинаковых заданий. Также желательно, чтобы оценоч-

ный документ охватывал как можно больше проверяемого контента, т. е. чтобы задания, входящие в оценочный документ, различались по атрибутам, не входящим в маску. Эта проблема решается при помощи кластеризации каждого класса заданий.

Атрибуты заданий принимают, как правило, значения в шкале наименований, т. е. являются качественными, а не количественными. Следовательно, для кластеризации классов заданий необходимо применять алгоритмы кластеризации категориальных данных. Самым популярным, легко программируемым, быстрым и масштабируемым является алгоритм CLOPE [8; 9]. В данном алгоритме коэффициент отталкивания подбирается автоматически. Алгоритм обладает следующим свойством: чем больше коэффициент отталкивания, тем больше кластеров получится в результате. Для поиска коэффициента мы применяем метод деления отрезка пополам, подбор осуществляем до тех пор, пока не достигнем нужного числа кластеров. Ожидаемым числом кластеров является количество ключей класса в шаблоне оценочного материала с погрешностью 2.

Если применять CLOPE в оригинальном виде, то возникают следующие проблемы [10].

1. Маленькие кластеры, которые получаются после кластеризации. Задания из таких кластеров будут часто повторяться при формировании комплекта оценочных документов.

2. Некоторые атрибуты логичнее и проще задать в виде чисел. Например, размер ответа. Классический алгоритм CLOPE плохо работает с числовыми признаками.

Для решения первой проблемы можно либо удалять, либо объединять маленькие кластеры. В данной предметной области потеря оценочных материалов критична, поэтому был выбран второй вариант – объединение. Для процесса объединения используется функция вычисления стоимости добавления. Стоимость удаления при этом не учитывается. Для разных объемов данных граничное значение маленького кластера может меняться. Поэтому данный критерий вычисляется динамически. Сначала находится медиана по числу объектов во всех кластерах. Границей маленького кластера будет максимум между значением по умолчанию – 2 и $[median \times 75\%] + 1$.

Таким образом удалось достичь адаптивного вычисления критерия маленького кластера в зависимости от результата кластеризации. Слияние кластеров происходит итерационно. Перед началом процесса формируется несколько списков: $S[]$ – список маленьких кластеров, $C[]$ – список кластеров, с которыми происходит объединение. Изначально $C[]$ равен результату кластеризации. Далее запускаем алгоритм удаления маленьких кластеров:

1. Выбираем маленький кластер $S[i]$. Если он уже не удовлетворяет критерию, удаляем его из $S[]$, переходим к следующей итерации.
2. Удаляем выбранный кластер $S[i]$ из $C[]$.
3. Для каждого объекта из $S[i]$ находим наиболее подходящий кластер из $C[]$ с использованием функции стоимости. Переносим объекты в найденные кластеры.
4. Удаляем $S[i]$.

Данный алгоритм удаляет не каждый маленький кластер. Если маленький кластер перестал быть таковым в ходе предыдущих итераций, то он не будет удален.

Для решения второй проблемы необходимо понять, почему CLOPE начинает плохо работать, когда мы используем числовые атрибуты. Дело в том, что для алгоритма даже соседние числа являются абсолютно разными. Из-за этого число итераций сравнений растет, а объекты с похожими атрибутами воспринимаются как разные. Для решения данной проблемы числовые признаки в нашем алгоритме преобразуются в интервалы. Во время вычисления стоимости проверяется не равенство числовых атрибутов, а их принадлежность к одному интервалу. Встает вопрос о том, на какое количество интервалов разбивать атрибут. Для этого используется правило Стерджеса [11]:

$$num = [\log_2 N] + 1,$$

где N – количество уникальных значений атрибута.

Алгоритм генерации комплекта оценочных документов

Следующим этапом происходит генерация комплекта оценочных документов на основе результатов работы алгоритма кластеризации. Это итерационный этап. Для каждого задания в шаблоне выбирается случайный вопрос из случайного кластера класса.

На этом этапе возникает проблема дублирования данных между разными классами. Для ее решения выбор заданий из кластеров происходит при необходимости с элементами перебора доступных заданий. Рассмотрим данную проблему подробнее, на примере:

Кластер 1: Задание 1, Задание 2

Кластер 2: Задание 1, Задание 3

Кластер 3: Задание 2, Задание 3

Предположим, что выбор вопросов из кластеров будет происходить в порядке возрастания номера кластера. В кластере 1 выбираем вопрос 2, в кластере 2 выбираем вопрос 3. После этого у нас нет возможности выбрать вопрос из кластера 3, потому что все имеющиеся в нем вопросы уже были использованы на предыдущих шагах. Поэтому происходит изменение выбранного вопроса в предыдущих кластерах, начиная с последнего. Из кластера 2 выбираем вопрос 1, тогда у нас появляется возможность выбрать вопрос 3 из кластера 3. Таким образом, решается обозначенная выше проблема.

Для того чтобы в один оценочный документ не попадали задания из одного кластера, итерации проводятся не только по заданиям, но и во время выбора кластера. В начале итерации фиксируется список доступных кластеров. Среди доступных выбирается случайный. Выбранный кластер убирается из списка доступных кластеров до следующей итерации. Итерация заканчивается, когда заканчивается список доступных кластеров.

Алгоритм генерации комплекта оценочных документов:

1. Итерируем по заданному количеству вариантов оценочных документов. Повторяем алгоритм до тех пор, пока не будет сгенерировано нужно число вариантов.
2. Создаем список результирующих заданий для варианта $Q[]$. Запускаем генерацию варианта.
3. Если номер генерируемого задания в варианте больше, чем требовалось, то выход.
4. K – ключ класса для текущего задания. По ключу получаем следующий доступный набор заданий B для класса K .
5. $BNum$ – количество заданий в наборе B . Обходим в случайном порядке задания из B .
6. Если задание уже есть в $Q[]$, то берем следующее. Запоминаем выбранное задание, кладем его в $Q[i]$.
7. Переходим к выбору задания $i + 1$ на шаг 5.
8. Если на шаге 6 удалось выбрать задание $i + 1$, завершаем выбор задания i .
9. Иначе удаляем задание i из $Q[]$. Переходим к следующему заданию из набора B . Переходим на шаг 5.
10. Если обошли все $BNum$ заданий из набора B , значит, доступных заданий нет: все уже были ранее использованы на предыдущих итерациях. Завершаем шаг с неудачей, возвращаемся к выбору задания $i - 1$.

Данный подход автоматически обрабатывает ситуацию, когда количество кластеров не совпадает с нужным количеством заданий в шаблоне. В итоге получаем необходимое количество вариантов сгенерированных оценочных материалов.

Заключение

В статье описана программная система управления оценочными средствами преподавателя. Система сокращает время, которое преподаватель тратит на подготовку проверочных материалов, позволяя в полуавтоматическом режиме создавать комплект оценочных документов.

В системе реализована возможность создавать и управлять банками заданий. Каждое задание в банке обладает набором атрибутов, которые участвуют в процессе генерации оценочных документов. В системе также реализована возможность автоматически генерировать нужное количество вариантов оценочных документов. При этом алгоритм генерации комплекта оценочных документов работает так, что в один комплект, с одной стороны, входят максимально схожие по структуре варианты оценочных документов, а с другой – каждый оценочный документ в комплекте уникален.

В основе алгоритма генерации комплекта оценочных документов лежит алгоритм кластеризации категориальных данных. В рамках данной работы была разработана модификация алгоритма CLOPE, позволяющая автоматически определять необходимое число кластеров, в зависимости от входных данных. Также данная модификация решает проблему маленьких кластеров и проблему категориальной кластеризации числовых атрибутов.

Список литературы

1. Долгих М. В. Формирование фонда оценочных средств как необходимое условие реализации основной профессиональной образовательной программы // Вестник Южно-Уральского профессионального института. 2014. № 1 (13). С. 36–45.
2. Sharp V. F. Computer Education for Teachers: Integrating Technology into Classroom Teaching. John Wiley & Sons, 2008, 416 p.
3. Анисимов А. М. Работа в системе дистанционного обучения Moodle: Учеб. пособие. 2-е изд. Харьков: ХНАГХ, 2009. 292 с.
4. Найданов Ч. А., Пальчунов Д. Е., Сазонова П. А. Теоретико-модельные методы интеграции знаний, извлечённых из медицинских документов // Вестник НГУ. Серия: Информационные технологии. 2015. Т. 13, № 3, С. 29–41.
5. Palchunov D., Yakhyayeva G., Dolgusheva E. Conceptual Methods for Identifying Needs of Mobile Network Subscribers. In: CEUR Workshop Proceedings, vol. 1624, p. 147–160.
6. Яхьяева Г. Э., Абсайдулueva А. Р. Семантический подход к моделированию фонда оценочных средств // Вестник НГУ. Серия: Информационные технологии. 2018. Т. 16, № 2. С. 113–121.
7. Баталин К. В., Мамеев Н. С., Попова К. Ю., Рыжаков И. Д., Яхьяева Г. Э. Программная система управления образовательным процессом ИТОС // Вестник НГУ. Серия: Информационные технологии. 2018. Т. 16, № 4. С. 20–30.
8. Yang Y., Guan H., You J. CLOPE: A fast and Effective Clustering Algorithm for Transactional Data In: Proc. of SIGKDD'02. Edmonton, Alberta, Canada, 2002.
9. Паклин Н. Б. Кластеризация категориальных данных: масштабируемый алгоритм CLOPE. Научная библиотека BaseGroup Labs. URL: <http://www.basegroup.ru/library/analysis/clusterization/clope>.
10. Баталин К. В. Оптимизация метода кластеризации категориальных данных базы оценочных материалов // Материалы Всерос. конференции с международным участием «Знания – Онтологии – Теории» (ЗОНТ-2019). Новосибирск, 2019. С. 414.
11. Шорохов И. С., Кисляк Н. В., Мариев О. С. Статистические методы анализа: Учеб. пособие. Екатеринбург: Изд-во Урал. ун-та, 2015. 300 с.

References

1. **Dolgikh M. V.** Formation of the Fund of evaluation funds as a necessary condition for the implementation of the main professional educational program. *Bulletin of the South Ural professional Institute*, 2014, no. 1 (13), p. 36–45. (in Russ.)
2. **Sharp V. F.** Computer Education for Teachers: Integrating Technology into Classroom Teaching. John Wiley & Sons, 2008, 416 p.
3. **Anisimov M. A.** Work in system of distance learning Moodle. Textbook. 2nd ed. Kharkov, KhNAGKh Press, 2009, 292 p. (in Russ.)
4. **Naidanov C. A., Palchunov D. E., Sazonov P. A.** Model-Theoretical methods for integration of knowledge extracted from medical records. *Vestnik NSU. Series: Information Technologies*, vol. 13, no. 3, 2015, p. 29–41. (in Russ.)
5. **Palchunov D., Yakhyaeva G., Dolgusheva E.** Conceptual Methods for Identifying Needs of Mobile Network Subscribers. In: CEUR Workshop Proceedings, vol. 1624, p. 147–160.
6. **Yakhyayeva G. E., Absaidulieva A. R.** Semantic approach to the modeling of the Foundation assessment tools. *Vestnik NSU. Series: Information Technologies*, 2018, vol. 16, no. 2, p. 113–121. (in Russ.)
7. **Batalin K. V., Mameev N. S., Popova K. Yu., Ryzhakov I. D., Yakhyaeva G. E.** Software system of educational process management ITOS. *Vestnik NSU. Series: Information Technologies*, 2018, vol. 16, no. 4, p. 20–30. (in Russ.)
8. **Yang Y., Guan H., You J.** CLOPE: A fast and Effective Clustering Algorithm for Transactional Data In: Proc. of SIGKDD'02. Edmonton, Alberta, Canada, 2002.
9. **Paklin N. B.** Clusterization of categorical data: a scalable SLOPE algorithm. BaseGroup Labs scientific library. URL: <http://www.basegroup.ru/library/analysis/clusterization/clope>. (in Russ.)
10. **Batalin K. V.** Optimization of the clustering method of categorical data of the evaluation materials database. In: Proceedings of the All-Russian conference with international participation “Knowledge – Ontology – Theory” (UMBRELLA-2019). Novosibirsk, 2019, p. 414. (in Russ.)
11. **Shorokhov I. S., Kislyak N. V., Mariev O. S.** Statistical methods of analysis. Studies guide. Ekanerinburg, Ural State Uni. Press, 2015, 300 p. (in Russ.)

Материал поступил в редколлегию
Received
10.05.2020

Сведения об авторах Information about the Authors

Баталин Кирилл Вячеславович, магистрант ФИТ Новосибирского государственного университета (Новосибирск, Россия)
k.v.batalin@gmail.com

Яхьяева Гульнара Эркиновна, кандидат физико-математических наук, доцент, доцент кафедры общей информатики ФИТ Новосибирского государственного университета (Новосибирск, Россия)
gul_nara@mail.ru

Information about the Authors

Kirill V. Batalin, Master of Science, Novosibirsk State University (Novosibirsk, Russian Federation)
k.v.batalin@gmail.com

Gulnara E. Yakhyaeva, Associate Professor of the Department of Information Technology, Novosibirsk State University (Novosibirsk, Russian Federation)
gul_nara@mail.ru