



Интерактивные тренажеры и их применение в процессе подготовки ИТ-специалистов в системе дистанционного обучения Moodle

Алексей Владимирович САМОХВАЛОВ ^{*}, Денис Сергеевич СОЛОВЬЕВ 

Александр Александрович СКВОРЦОВ 

ФГБОУ ВО «Тамбовский государственный университет им. Г.Р. Державина»
392000, Российская Федерация, г. Тамбов, ул. Интернациональная, 33

*Адрес для переписки: samohvalov@gmail.com

Актуальность. Рассмотрено применение интерактивных тренажеров в подготовке ИТ-специалистов. Актуальность обусловлена необходимостью качественной подготовки специалистов в области информационных технологий для цифровой экономики страны в условиях существенного дефицита кадров, высокой динамики знаний изучаемой предметной области и необходимости постоянной связи обучения с включением студентов в реальную практико-ориентированную профессиональную деятельность. Разработан модуль автоматизированной проверки программного кода для СДО Moodle и с его использованием созданы интерактивные тренажеры для подготовки специалистов в области веб-технологий.

Методы исследования. Для достижения поставленных целей исследования применялись различные методы, включая анализ педагогической литературы, обобщение и систематизацию данных. При разработке интерактивного тренажера использовались различные методы и подходы, а именно: структурно-функциональное моделирование для описания устройства тренажера; принципы объектно-ориентированного программирования для обеспечения гибкости и расширяемости тренажера; интеграция в систему дистанционного обучения для использования тренажера в рамках обучающего курса; статистическая обработка результатов эксперимента для анализа эффективности тренажера и внесения необходимых коррективов.

Результаты исследования. Изменения в научно-экономической сфере нашей страны усилили противоречия между потребностями общества и уровнем ИТ-специалистов, производимых системой подготовки кадров. Технологическая модернизация, интеграция научных областей и постоянное формирование новых знаний диктуют новые требования к ИТ-специалистам. От хорошего образовательного курса подготовки ИТ-специалистов мы ожидаем не только большого объема практических заданий, но и возможности быстро получать обратную связь. Онлайн-тренажеры успешно применяются многими образовательными платформами для интенсификации процесса подготовки ИТ-специалистов, повышения вовлеченности студентов в образовательный процесс. Был разработан модуль автоматизированной проверки HTML и CSS кода для СДО Moodle и созданы на его основе онлайн-тренажеры. Проведен эксперимент по внедрению разработанных тренажеров в учебный процесс. Студенты, исходя из полученных заданий, могут самостоятельно писать код, работать с ошибками, брать подсказки, получать немедленную обратную связь и переходить к следующим заданиям. Для оценки эффективности использования разработанных тренаже-

ров в учебном процессе было выбрано соотношение числа студентов, успешно выполнивших практические задания, и тех, кто к ним приступал. Этот показатель мы сравнили с показателем успешного выполнения практических заданий в группах, где тренажера не было. Для эксперимента были отобраны группы студентов со сравнимым исходным уровнем компетенций в области веб-разработки. Проведенный эксперимент показал, что в группах, проходящих обучение с использованием разработанных интерактивных тренажеров, соотношение успешно выполненных практических заданий на 18 % выше, чем в группах студентов, обучающихся без использования тренажеров.

Выводы. Эффективный выбор имеющихся инструментальных ресурсов позволяет быстро и экономично внедрять компоненты дистанционного обучения, специфичные для ИТ-специальностей, в соответствии с потребностями цифровой экономики Российской Федерации. Использование таких технологических инструментов не только повышает качество образования, но и готовит студентов к запросам современного рынка труда. Высшим учебным заведениям крайне важно адаптироваться к меняющимся потребностям цифровой экономики, а применение интерактивных тренажеров является одним из ключевых шагов в данном направлении. Разработанный модуль автоматизированной проверки HTML и CSS кода для СДО Moodle и построенных на его основе тренажеров позволит усилить практическую составляющую подготовки специалистов по веб-технологиям в системе высшего образования. Это обеспечит приобретение студентам практических навыков и знаний, необходимых для их будущей карьеры в быстро развивающемся цифровом мире.

Ключевые слова: интерактивные тренажеры, система открытого образования, дистанционное обучение, подготовка ИТ-специалистов, цифровизация образования

Для цитирования: Самохвалов А.В., Соловьев Д.С., Скворцов А.А. Интерактивные тренажеры и их применение в процессе подготовки ИТ-специалистов в системе дистанционного обучения Moodle // Вестник Тамбовского университета. Серия: Гуманитарные науки. 2023. Т. 28. № 5. С. 1063-1076. <https://doi.org/10.20310/1810-0201-2023-28-5-1063-1076>

Original article

<https://doi.org/10.20310/1810-0201-2023-28-5-1063-1076>

Interactive simulators and their application in the process of training IT specialists on the Moodle learning management system

Alexey V. SAMOKHVALOV *, Denis S. SOLOVJEV , Alexander A. SKVORTSOV 

Derzhavin Tambov State University

33 Internatsionalnaya St., Tambov, 392000, Russian Federation

*Corresponding author: samohvalov@gmail.com

Importance. The use of interactive simulators in the training of IT specialists is considered. The relevance is due to the need for high-quality training of specialists in the field of information technology for the country's digital economy in conditions of a significant shortage of personnel, high dynamics of knowledge in the subject area being studied and the need for a constant connection between training and the inclusion of students in real practice-oriented professional activities. A module for automated verification of program code for the Moodle LMS is developed and, using it, interactive simulators have been created for training specialists in the field of web technologies.

Research Methods. To achieve the research goals, various methods are used, including analysis of pedagogical literature, generalization and systematization of data. When developing an in-

teractive simulator, various methods and approaches are used, namely: structural and functional modeling to describe the design of the simulator; principles of object-oriented programming to ensure flexibility and extensibility of the simulator; integration into a distance learning system for using the simulator as part of a training course; statistical processing of experimental results to analyze the effectiveness of the simulator and make the necessary adjustments.

Results and Discussion. Changes in the scientific and economic sphere of our country have increased the contradictions between the needs of society and the level of IT specialists produced by the personnel training system. Technological modernization, integration of scientific fields and the constant formation of new knowledge dictate new requirements for IT specialists. From a good educational course for training IT specialists, we expect not only a large volume of practical tasks, but also the ability to quickly receive feedback. Online simulators are successfully used by many educational platforms to intensify the training process of IT specialists and increase student involvement in the educational process. A module for automated checking of HTML and CSS code for the Moodle LMS is developed and online simulators were created based on it. An experiment is conducted to introduce the developed simulators into the educational process. Students, based on the assignments they receive, can write code on their own, work with errors, take hints, receive immediate feedback, and move on to the next assignments. To assess the effectiveness of using the developed simulators in the educational process, the ratio of the number of students who successfully completed practical tasks and those who started them is chosen. We compared this indicator with the indicator of successful completion of practical tasks in groups where there was no simulator. For the experiment, groups of students with a comparable initial level of competencies in the field of web development are selected. The experiment showed that in groups undergoing training using the developed interactive simulators, the ratio of successfully completed practical tasks is 18 % higher than in groups of students studying without the use of simulators.

Conclusion. Effective selection of available instrumental resources allows you to quickly and economically implement distance learning components specific to IT specialties, in accordance with the needs of the digital economy of the Russian Federation. The use of such technological tools not only improves the quality of education, but also prepares students for the demands of the modern labor market. It is extremely important for higher education institutions to adapt to the changing needs of the digital economy, and the use of interactive simulators is one of the key steps in this direction. The developed module for automated checking of HTML and CSS code for the Moodle LMS and simulators built on its basis will strengthen the practical component of training specialists in web technologies in the higher education system. This will ensure that students acquire the practical skills and knowledge needed for their future careers in the rapidly evolving digital world.

Keywords: interactive simulators, open education system, distance learning, IT specialists training, education digitalization

For citation: Samokhvalov, A.V., Solovjev, D.S., & Skvortsov, A.A. (2023). Interactive simulators and their application in the process of training IT specialists on the Moodle learning management system. *Vestnik Tambovskogo universiteta. Seriya: Gumanitarnye nauki = Tambov University Review. Series: Humanities*, vol. 28, no. 5, pp. 1063-1076. (In Russ., abstract in Eng.) <https://doi.org/10.20310/1810-0201-2023-28-5-1063-1076>

АКТУАЛЬНОСТЬ

Изменения в научной и экономической сферах нашей страны обострили противоречия между потребностями общества и уровнем ИТ-специалистов, подготавливаемых системой высшего образования. Технологиче-

ская модернизация, интеграция научных областей и постоянное формирование новых знаний диктуют новые требования к ИТ-специалистам.

В настоящее время наблюдается значительный дефицит ИТ-кадров как в России, так и в мире. По оценкам Автономной не-

коммерческой организации «Цифровая экономика» к 2025 г. спрос на них увеличится до 190 млн рабочих мест¹.

Ассоциация предприятий компьютерных и информационных технологий прогнозирует, что к 2024 г. спрос на высококвалифицированных специалистов в сфере ИКТ в России увеличится до 290–300 тыс. человек в год². При этом регионы России будут испытывать большой дефицит кадров, так как большая доля специалистов сосредоточена в Москве (около 20 %). Таким образом, особо острую потребность в ИТ-кадрах испытывает цифровая экономика регионов России.

К ИТ-специалистам высшего уровня квалификации в России относят следующие группы специальностей³: разработчики программного обеспечения; разработчики веб- и мультимедийных приложений; системные аналитики; системные администраторы; специалисты по компьютерным сетям и др.

Подготовка ИТ-специалистов в открытой системе образования имеет свои особенности, которые определяются высокой динамикой знаний в предметной области и необходимостью постоянной связи обучения с привлечением студентов к реальной практико-ориентированной профессиональной деятельности [1; 2]. К настоящему времени был проведен ряд исследований, раскрывающих методологические и технологические аспекты дистанционного обучения. Данной проблеме посвящены работы Е.С. Полат, А.А. Андреева, И.В. Роберт, А.Н. Сергеева, В.П. Тихомирова, М.В. Храмовой, А.В. Хуторского, С.А. Щенникова, А.А. Скворцова и мн. др. Однако для эффективной подготовки

ИТ-специалистов в системе открытого образования необходимо учитывать особенности данной предметной области и обеспечивать постоянную связь между обучением и реальной профессиональной деятельностью студентов.

Проблема построения адаптивных образовательных платформ для системы открытого образования рассматривается в работах М.В. Храмовой, S. Drissi, A. Abdelkrim, R.M. Bernard, E. Borokhovski [3–5].

Проблемам разработки компьютерных тренажеров и их применению в системе открытого образования посвящены работы Н.Е. Артес, В.К. Маркелова, Н.Н. Костиной и др. [6–8].

Ожидается, что хороший образовательный курс для ИТ-специалистов будет иметь не только большой объем практических задач, но и возможность быстро получать обратную связь. Нельзя игнорировать человеческий фактор: как бы ни старались оперативно преподаватели проверять задания, все равно данный процесс требует значительного времени. Поскольку скорость обработки заданий влияет на результаты студентов, процесс проверки выполненных заданий необходимо оптимизировать. Еще одной важной задачей является сокращение рабочей нагрузки преподавателей, с тем чтобы они могли сосредоточиться на предоставлении качественных заданий.

Практика позволяет студентам применить свои теоретические знания. Однако некоторые базовые навыки требуют выполнения большого количества заданий. Это означает, что преподаватели должны тратить часы на проверку повторяющихся заданий. Интерактивные тренажеры [1; 2] помогают решить проблемы своевременной оценки выполненных заданий и предоставления рекомендаций студентам, а также снижения нагрузки преподавателей в системах открытого образования.

Таким образом, задачи, которые решают тренажеры, включают в себя:

– сокращение количества плановых проверок заданий педагогами;

¹ Автономная некоммерческая организация «Цифровая экономика» рассказала о кадрах в ИТ на Russian Internet Week 2022. URL: <https://data-economy.ru/news/tpost/4u5dds0bz1-ano-tsifrovaya-ekonomika-rasskazala-o-ka>

² ИТ-кадры для цифровой экономики в России. Оценка численности ИТ-специалистов в России и прогноз потребности в них до 2024 г. https://apkit.ru/files/it-personnel_research_2024_APKIT.pdf

³ Общероссийский классификатор занятий (принят и введен в действие Приказом Росстандарта от 12.12.2014 № 2020-ст) (ред. от 18.02.2021).

– повышение эффективности образовательного процесса.

В свою очередь, студенты также ожидают мгновенной обратной связи. Это означает получение быстрого ответа на их выполненное задание, который позволит им двигаться дальше. Это сохраняет мотивацию и интерес к дальнейшему обучению.

Преимущества тренажеров в сравнении с традиционными методами обучения:

- высокая скорость обучения;
- точность и контроль;
- высокий эффект интерактивности и геймификации;
- возможность многократного повторения упражнения.

Онлайн-тренажеры успешно применяются многими образовательными платформами – HTML Academy, Яндекс Практикум, Нетология, W3Schools, Stepik и др. Используемая в Тамбовском государственном университете им. Г.Р. Державина СДО Moodle имеет возможность подключения дополнительных модулей. Имеются внешние модули проверки программного кода (наиболее известный – CodeRunner), но для автоматизированной проверки HTML и CSS кода (что необходимо для подготовки специалистов в области веб-технологий) они не пригодны.

Именно поэтому нами был разработан модуль автоматизированной проверки HTML и CSS кода для СДО Moodle и созданы на его основе онлайн-тренажеры. Тренажеры применяются для обучения основам веб-разработки. Студенты, исходя из полученных заданий, могут самостоятельно писать код, работать с ошибками, брать подсказки, получать мгновенную обратную связь и переходить к следующим заданиям.

МЕТОДЫ ИССЛЕДОВАНИЯ

В работе использовались метод анализа источников информации, взятых из журналов, индексированных в Web of Science и Scopus, публикации из научных электронных библиотек открытого доступа, таких как

eLibrary и CyberLeninka, поисковая система Google Scholar.

При разработке интерактивного тренажера использовались различные методы и подходы, а именно: структурно-функциональное моделирование для описания устройства тренажера; принципы объектно-ориентированного программирования для обеспечения гибкости и расширяемости тренажера; статистическая обработка результатов эксперимента для анализа эффективности тренажера и внесения необходимых корректив.

РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЯ

Разработан модуль автоматизированной проверки HTML и CSS кода для СДО Moodle. На его основе созданы тренажеры для подготовки ИТ-специалистов в области веб-технологий с возможностью оценки и самооценки.

Схема взаимодействия пользователя и интерактивного тренажера в системе дистанционного обучения Moodle представлена на рис. 1 и предполагает два типа пользователей: преподаватель и студент.

Тренажер состоит из трех основных элементов: форма создания задания, форма выполнения задания и модуль проверки.

Преподаватель отвечает за загрузку данных задания и соответствующих правил на сервер через форму создания задания. Эти данные и правила затем сохраняются на сервере и могут быть доступны студенту.

Студент, с другой стороны, взаимодействует с тренажером через форму выполнения задания. Эта форма позволяет ему предоставить свой ответ на задание. Далее форма выполнения задания отправляет ответ студента и задание в модуль проверки.

Модуль проверки проверяет код на соответствие заранее определенным правилам, загруженным преподавателем, и предоставляет результат исполнителю задачи. В свою очередь, форма выполнения задачи возвращает студенту список правил и результат

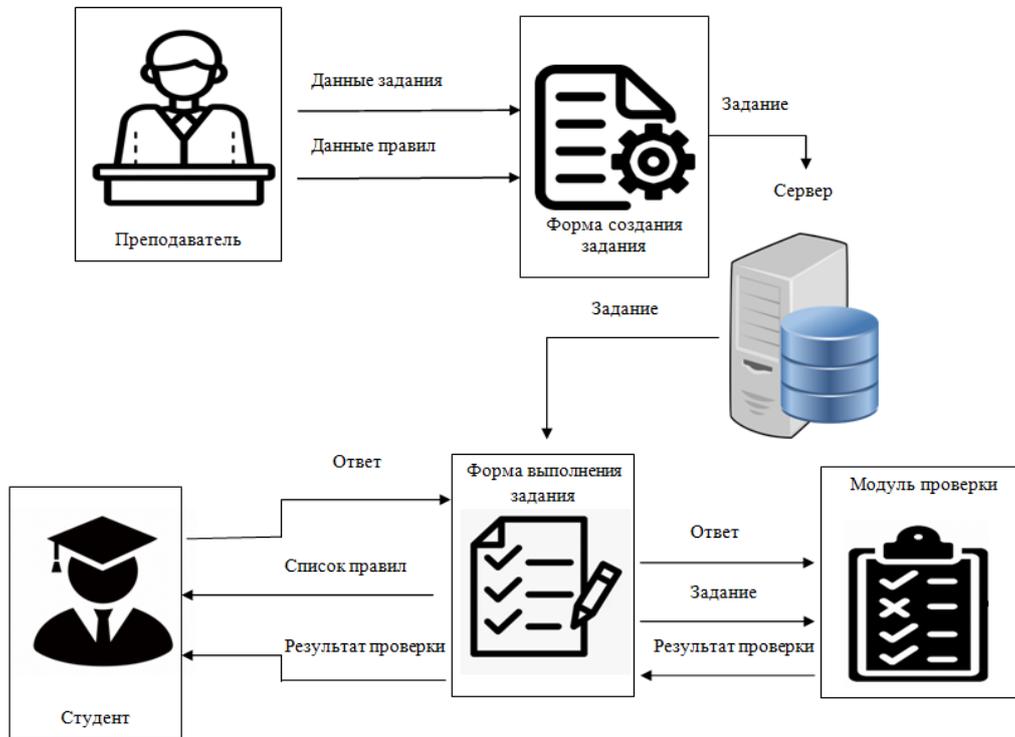


Рис. 1. Схема взаимодействия пользователя с интерактивным тренажером в системе дистанционного обучения Moodle

Fig. 1. Scheme of user interaction with an interactive simulator in the Moodle learning management system

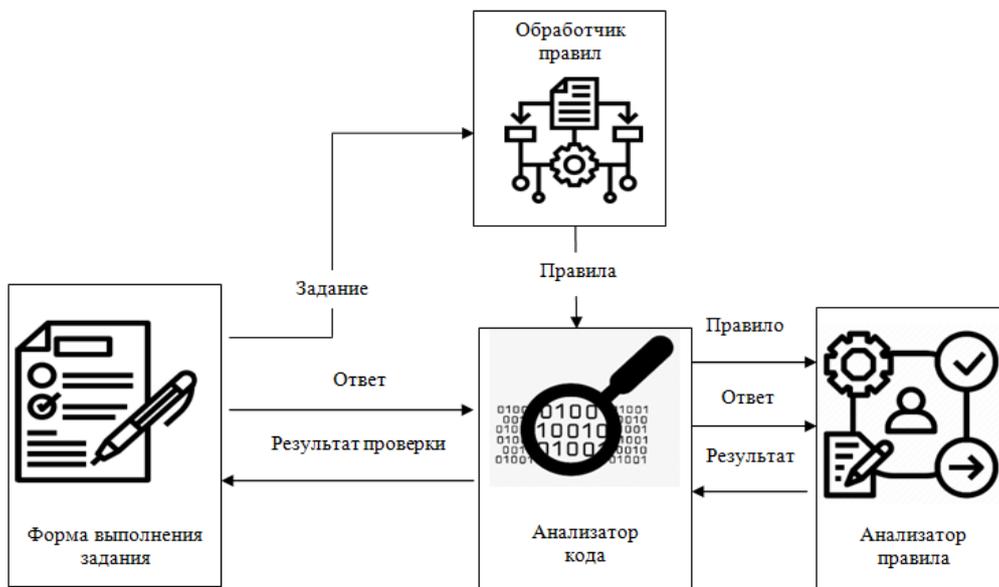


Рис. 2. Схема взаимодействия компонентов модуля

Fig. 2. Interaction diagram of module components

проверки. Это позволяет студенту понять свои ошибки и улучшить понимание задачи.

Предложенная схема взаимодействия обеспечивает бесперебойную связь между преподавателем и студентом, реализуя эффективное дистанционное обучение через интерактивный тренажер.

Рассмотрим подробнее состав модуля проверки, состоящего из нескольких компонентов, которые взаимодействуют между собой (рис. 2): форма выполнения задания, обработчик правил, анализатор кода и анализатор правила.

Форма выполнения задания реализует интерфейс, через который пользователь может ввести свой код и отправить его на проверку. Форма передает код обработчику правил, который его проверяет. Обработчик получает данные от анализатора правил и применяет их к полученному коду. Полученный результат возвращается анализатору кода.

Анализатор является основным компонентом модуля и отвечает за проверку правильности кода. Он получает код от обработчика правил и выполняет его по шагам, применяя к каждому шагу соответствующее правило. Если все шаги выполнены правильно, то анализатор кода возвращает положительный результат обратно в форму выполнения задания. В противном случае, если хотя бы один шаг выполнен неправильно, то анализатор кода возвращает отрицательный результат с указанием локализации ошибки.

Анализатор правила получает информацию о типе задания и определяет, какие правила должны быть применены к коду.

Таким образом, модуль проверки правильности кода работает следующим образом: форма выполнения задания передает код обработчику правил, который получает правила от анализатора правил и применяет их к коду. Результат обработки передается анализатору кода, который выполняет код по шагам и применяет к каждому шагу правило, полученное от анализатора правила. Если все шаги выполнены правильно, то возвращается положительный результат, иначе – отрицательный с указанием локализации ошибки.

Разработанный модуль проверки позволяет эффективно оценивать правильность кода и дает возможность пользователю-студенту исправлять ошибки и улучшать свои навыки программирования.

Фрагмент алгоритма работы модуля проверки правильности кода представлен в листинге 1.

Листинг 1. Фрагмент алгоритма работы модуля проверки правильности кода
МодульПроверки(задание, ответ):

```
{
    необработанные_правила=задание.правила
    для каждого правила в необработанные_правила:
    {
        обработанное_правило=обработатьПравило(правило,
        ответ)
        список_правил.добавить(обработанное_правило)
    }
    для каждого правила в список_правил:
    {
        результат_проверки.добавить(проверитьПравило(правило))
    }
    вернуть результат_проверки
}
```

Модуль представляет собой набор функций, предназначенных для проверки правильности выполнения кода, который был написан пользователем-студентом. Входными данными для модуля являются задание и ответ, которые передаются в качестве параметров. Первым шагом модуль сохраняет правила задания в переменную «необработанные_правила». Далее, с помощью цикла для каждого правила в необработанных правилах вызывается функция «обработатьПравило», которая принимает на вход правило и ответ пользователя и возвращает обработанное правило. Обработанное правило добавляется в «список_правил».

После того как все правила были обработаны, модуль запускает цикл для каждого правила в «список_правил». Внутри цикла вызывается функция «проверитьПравило», которая принимает на вход правило и сравнивает его с ответом пользователя. Результат проверки добавляется в переменную «результат_проверки».

В конце работы модуля результат проверки возвращается как результат выполнения функции. Таким образом, модуль про-

верки правильности кода позволяет автоматически проверить правильность выполнения задания пользователем, сравнивая его ответ с правилами задания. Это упрощает процесс проверки и позволяет быстро выявить ошибки в коде. Также модуль может быть использован как инструмент для обучения программированию, позволяя пользователям самостоятельно проверять свой код и исправлять ошибки. В будущем модуль может быть доработан для поддержки различных языков программирования и расширенных функций проверки, таких как анализ сложности кода и оптимизация решений.

Были созданы функции добавления (`htmleditor_add_instance`), изменения (`htmleditor_update_instance`) и удаления (`htmleditor_delete_instance`) данных правил. Рассмотрим принцип работы данных функций более подробно.

Функция `htmleditor_add_instance` отвечает за добавление нового экземпляра редактора кода в систему Moodle. Он использует объект `htmleditor` в качестве параметра и глобальные переменные для доступа к базе данных. Функция `insert_record` используется для вставки объекта `htmleditor` в таблицу `htmleditor` в базе данных. Устанавливается идентификатор объекта `htmleditor` равным идентификатору возврата для вновь вставленной записи.

Функция `htmleditor_update_instance` отвечает за обновление экземпляра редактора кода в системе Moodle. Она принимает объект `htmleditor` в качестве параметра. Затем извлекает все переменные объекта `htmleditor` и сохраняет их в массиве. Далее выполняет циклический поиск каждой переменной и проверяет, содержит ли она определенные ключевые слова. В этом случае необходимая информация сохраняется в массиве правил. Этот массив содержит всю информацию о правилах, связанных с редактором кода. Затем функция проходит через ключи массива правил и обновляет соответствующие поля в таблице `htmleditor_rules` в базе данных.

Функция `htmleditor_delete_instance` принимает идентификатор в качестве параметра и удаляет соответствующий экземпляр из базы данных в системе Moodle. Если экземпляр не найден, функция возвращает значение `false`. Далее функция извлекает правила, связанные с экземпляром, и сохраняет их в строковой переменной. Если существует несколько правил, они разделяются. Затем функция выполняет итерацию по каждому идентификатору правила в массиве и использует функцию `get_records` для извлечения всех записей из таблицы `htmleditor_rules`, соответствующих заданному идентификатору правила. Это гарантирует, что все связанные правила также будут удалены. Наконец, функция удаляет экземпляр из таблицы `htmleditor` в базе данных.

Далее были созданы функции для редактора правил: анализа правила (`analyze`), получения правила (`getRules`) и обновления просмотра (`UpdateAndCheck`). Рассмотрим принцип работы данных функций более подробно.

Функция `analyze` принимает четыре параметра: `htmlValue`, `cssValue`, `rules_unparsed` и `description`. Сначала инициализируется пустой массив `rules`. Затем разбирается параметр `rules_unparsed` в объект JSON и назначается массиву `rules`. Выполняется итерация в каждом правиле в массиве `prepared_rules` и вызывается функция `validRule` для параметра `htmlValue` и текущего правила. Результат этого вызова функции назначается логической переменной `is_valid`. Функция регистрирует выполнение правила, обновляя переменную `all_valid`, и вызывает функцию `getDescription` для текущего правила и переменную `is_valid`. Возвращаемое значение добавляется в строку `description_text`. После анализа всех правил функция устанавливает источник параметра `description` в новом URL, который кодирует строку `description_text` как UTF-8. Наконец, функция возвращает значение переменной `all_valid`.

Программная реализация функции `analyze` представлена ниже.

```
function analyze(htmlValue, cssValue, rules_unparsed,
description) {
  let rules = [];
  try {rules = JSON.parse(rules_unparsed);}
  catch (e) {
    dbm(['e', 'rule:', rules_unparsed], 'err');
  }
  let description_text = "";
  let all_valid = true;
  let prepared_rules = prepareRules(rules);
  dbm([prepared_rules]);
  prepared_rules.forEach(prepared_rule => {
    dbm(['analyze rule', prepared_rule]);
    let is_valid = validRule(htmlValue, prepared_rule);
    dbm(['rule is valid:', Boolean(is_valid)], 'warn');
    all_valid &= is_valid;
    description_text += getDescription(prepared_rule,
is_valid);
  })
  description.src = "data:text/html;charset=UTF-8,"
+ encodeURIComponent(description_text);
  return all_valid;
}
```

Функция `getRules` принимает идентификатор задачи и возвращает строку правил, связанных с ней. Правила, связанные с задачей, сохраняются в массиве `rules_with_childs`. Правила разделяются на отдельные элементы с помощью функции `explode` и сохраняются в `rules_array`. Затем каждое правило передается функции `getRulesWireChilds`, которая добавляет все дочерние правила к исходному правилу и возвращает их. Эти обновленные правила далее добавляются в массив `rules_with_childs`.

Функция `UpdateAndCheck` выполняет два основных действия. Во-первых, она извлекает текущий код из редакторов HTML и CSS и объединяет его в одну строку с кодом CSS, заключенным в теги `<style>`. Этот комбинированный код затем назначается атрибуту `<src>` элемента предварительного просмотра, который отображается на странице. Далее вызывается функция `analyze`, передавая код из редакторов HTML и CSS, а также набор правил и описаний. Функция `analyze` проверяет код на соответствие заданным правилам и возвращает логическое значение, указывающее на то, прошел ли код анализ или нет.

Реализация рассмотренных функций редактора правил воплощена на форме, интерфейс которой показан на рис. 3.

Данная форма позволяет добавлять, удалять и редактировать задания. Каждое задание состоит из правил, каждое правило может иметь дочерние правила, при этом проверка дочерних правил будет производиться внутри родительских элементов. Так же правила, которые не должны быть отображены, необходимо сделать невидимыми, для этого у каждого правила есть специальная кнопка «Сделать скрытым».

Рассмотрим форму создания правил (см. рис. 4).

Обозначения на рис. 4 соответствуют пунктам:

1) поле для имени правила используется для краткого описания правила;

2) поле текста правила. При вводе необходимо придерживаться следующих правил:

а) текст должен представлять из себя HTML-тег либо обычный текст.

б) текст правила должен быть составлен в соответствии со всеми правилами написания HTML-тегов. Некорректное написание тега приведет к ошибке во время проверки.

3) поле для расширенного описания задания используется для более подробного описания того, что требуется от пользователя для выполнения правила;

4) кнопка «Сделать скрытым» / «Сделать отображаемым» отвечает за видимость правила в списке правил в редакторе. Рекомендуется иметь хотя бы одно отображаемое задание;

5) кнопка «Добавить правило», находящаяся в пределах рамки правила, добавляет дочерние правило. Дочернее правило отличается тем, что требует для своего выполнения нахождения внутри родительского тега. Если правило не имеет родителя, поиск правила будет происходить внутри тега `<<body>>`, `<<head>>` и на самом высоком уровне;

6) кнопка «Удалить правило» удаляет правило и все его дочерние правила;

Имя задания !

Начальное значение !

Имя задания 1 !

Текст правила !

Описание !

Сделать скрытым

Добавить задание Удалить задание присутствие ▾

Добавить задание

Рис. 3. Форма создания заданий
Fig. 3. Task creation form

Имя правила 1

Текст правила 2

Описание 3

Сделать скрытым 4

Добавить правило 5 Удалить правило 6 присутствие 7 ▾

Текст правила !

Сделать отображаемым

Добавить правило 8 Удалить правило присутствие 9 ▾

Добавить правило

Рис. 4. Форма создания правил
Fig. 4. Rule creation form



Рис. 5. Пример работы созданного тренажера
Fig. 5. Example of the created simulator

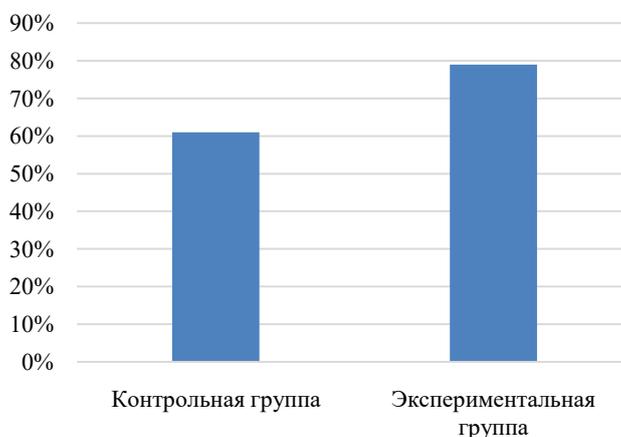


Рис. 6. Процент успешно выполненных практических заданий в экспериментальной и контрольной группах

Fig. 6. Percentage of successfully completed practical tasks in the experimental and control groups

7) поле выбора типа правила может находиться в состоянии «присутствие» либо «отсутствие». При проверке будет проверяться наличие либо отсутствие правила со-

ответственно. Важно отметить, что правило с типом «отсутствие» не должно иметь дочерних правил с типом «присутствие», иначе правило не сможет быть выполнено;

8) дочернее правило представляет из себя такое же правило, но для его выполнения необходимо, кроме соблюдения текста правила, нахождение правила внутри родительского тега;

9) кнопка «Добавить правило», находящаяся вне других правил, добавляет новое правило без родителя. Поиск такого правила будет происходить внутри тега «<body>», «<head>» и на самом высоком уровне.

Один из примеров работы созданного тренажера представлен на рис. 5.

С целью проверки эффективности разработанных тренажеров было проведено экспериментальное исследование. Для эксперимента были отобраны группы студентов с сопоставимыми исходными компетенциями в веб-разработке. В эксперименте приняли участие 42 студента направления 09.03.03 «Прикладная информатика» Тамбовского государственного университета им. Г.Р. Державина.

Для оценки эффективности использования разработанных тренажеров в учебном процессе было выбрано соотношение числа студентов, успешно выполнивших практические задания, к тем, кто к ним приступал, в качестве целевого показателя. Этот показатель в экспериментальной группе сравнивали с показателем успешного выполнения практических задач в группах, где тренажеры не использовались (контрольная группа). Полученные результаты показаны на рис. 6.

Проведенный эксперимент показал, что в группах, проходящих обучение с использо-

ванием разработанных интерактивных тренажеров, процент успешно выполненных практических заданий на 18 % выше, чем в группах студентов, обучающихся без использования тренажеров. Это свидетельствует о высокой эффективности разработанных тренажеров в обучении студентов веб-разработке.

ВЫВОДЫ

Эффективный выбор имеющихся инструментальных ресурсов позволяет быстро и экономично внедрять компоненты дистанционного обучения, специфичные для ИТ-специальностей, в соответствии с потребностями цифровой экономики РФ. Использование таких технологических инструментов не только повышает качество образования, но и готовит студентов к запросам современного рынка труда. Высшим учебным заведениям принципиально важно адаптироваться к меняющимся потребностям цифровой экономики, а использование интерактивных симуляторов – один из ключевых шагов в этом направлении. Разработанный модуль автоматизированной проверки HTML и CSS-кода для СДО Moodle и основанных на нем тренажеров позволит усилить практическую составляющую подготовки специалистов по веб-технологиям в системе высшего образования. Это даст студентам практические навыки и знания, необходимые для их будущей карьеры в быстро развивающемся цифровом мире.

Список источников

1. Лебедева Т.Ф., Муравьев С.А. Профессиональные стандарты в сфере информационных технологий как руководство к действию // Профессиональное образование в России и за рубежом. 2016. № 4 (24). С. 92-97. <https://elibrary.ru/xepjet>
2. Чванова М.С., Храмова М.В., Самохвалов А.В. [и др.] Особенности и выбор инструментария реализации системы дистанционного обучения для наукоемких специальностей // Психолого-педагогический журнал Гаудеамус. 2012. Т. 2. № 20. С. 51-59. <https://elibrary.ru/pgdxhf>
3. Храмова М.В., Александрова Н.А. Адаптивные платформы – как основной тренд систем дистанционного обучения в цифровую эпоху // Информационные технологии в образовании. 2020. № 3. С. 308-312. <https://elibrary.ru/lwimoh>

4. Sarnia D., Amirat A. An Adaptive E-Learning System based on Student's Learning Styles: An Empirical Study // *International Journal of Distance Education Technologies*. 2016. Vol. 14 (3). P. 34-51. <http://dx.doi.org/10.4018/IJDET.2016070103>
5. Bernard R.M. [et. al.] Twenty-first century adaptive teaching and individualized learning operationalized as specific blends of student-centered instructional events: A systematic review and metaanalysis // *Review of Educational Research*. 2015. Vol. 85. № 2. P. 275-314. <https://doi.org/10.1002/cl2.1017>
6. Артес Н.Е., Ростовский Н.С. Компьютерные тренажеры и их роль в системе современного образовательного процесса // *Аллея науки*. 2018. Т. 3. № 6 (22). С. 1031-1035. <https://elibrary.ru/xulobv>
7. Маркелов В.К., Завьялова О.А. Интерактивные задания в обучении программированию как инструмент учителя информатики // *Цифровая гуманитаристика и технологии в образовании (DHTE 2021): сб. ст. 2 Всерос. науч.-практ. конф. с междунар. участием. М.: МГППУ, 2021. С. 387-399. <https://elibrary.ru/bzgnqu>*
8. Костина Н.Н., Костин А.В., Минкин А.В. Компьютерные тренажеры для подготовки будущих учителей // *Ученые записки Крымского федерального университета имени В. И. Вернадского. Социология. Педагогика. Психология*. 2022. Т. 8 (74). С. 36-46. <https://elibrary.ru/eiqepc>

References

1. Lebedeva T.F., Murav'ev S.A. (2016). Professional standards in the IT sector as a guide to action. *Professional'noe obrazovanie v Rossii i za rubezhom = Professional Education in Russia and Abroad*, no. 4 (24), pp. 92-97. (In Russ.) <https://elibrary.ru/xepjet>
2. Chvanova M.S., Khramova M.V., Samokhvalov A.V. et al. (2012). Osobennosti i vybor instrumentariya realizatsii sistemy distantsionnogo obucheniya dlya naukoemkikh spetsial'nostei [Features and selection of tools for implementing a distance learning system for science-intensive specialties]. *Psikhologopedagogicheskii zhurnal Gaudeamus = Psychology-Pedagogical Journal Gaudeamus*, vol. 2, no. 20, pp. 51-59. (In Russ.) <https://elibrary.ru/pgdxhf>
3. Khramova M.V., Aleksandrova N.A. (2020). Adaptivnye platformy – kak osnovnoi trend sistem distantsionnogo obucheniya v tsifrovuyu epokhu [Adaptive platforms – as the main trend of distance learning systems in the digital era]. *Informatsionnye tekhnologii v obrazovanii* [Information Technologies in Education], no. 3, pp. 308-312. (In Russ.) <https://elibrary.ru/lwimoh>
4. Sarnia D., Amirat A. (2016). An adaptive e-learning system based on student's learning styles: an empirical study. *International Journal of Distance Education Technologies*, vol. 14 (3), pp. 34-51. <http://dx.doi.org/10.4018/IJDET.2016070103>
5. Bernard R.M. et al. (2015). Twenty-first century adaptive teaching and individualized learning operationalized as specific blends of student-centered instructional events: a systematic review and metaanalysis. *Review of Educational Research*, vol. 85, no. 2, pp. 275-314. <https://doi.org/10.1002/cl2.1017>
6. Artes N.E., Rostovskii N.S. (2018). Komp'yuternye trenazhery i ikh rol' v sisteme sovremennogo obrazovatel'nogo protsessa [Computer simulators and their role in the modern educational process]. *Alleya nauki* [Alley of Science], vol. 3, no. 6 (22), pp. 1031-1035. (In Russ.) <https://elibrary.ru/xulobv>
7. Markelov V.K., Zav'yalova O.A. (2021). Interaktivnye zadaniya v obuchenii programmirovaniyu kak instrumentarii uchitelya informatiki [Interactive tasks in teaching programming as a toolkit for computer science teachers]. *Sbornik statei 2 Vserossiiskoi nauchno-prakticheskoi konferentsii s mezhdunarodnym uchastiem «Tsifrovaya gumanitaristika i tekhnologii v obrazovanii» (DHTE 2021)* [Collection of Works of the 2nd All-Russian Scientific and Practical Conference with International Participation “Digital Humanities and Technology in Education”]. Moscow, Moscow State University of Psychology and Education Publ., pp. 387-399. (In Russ.) <https://elibrary.ru/bzgnqu>
8. Kostina N.N., Kostin A.V., Minkin A.V. (2022). Computer simulators for future teachers training. *Uchenye zapiski Krymskogo federal'nogo universiteta imeni V.I. Vernadskogo. Sociology. Pedagogy. Psychology* [Scientific Notes of the V.I. Vernadsky Crimean Federal University. Sociology. Pedagogy. Psychology], vol. 8 (74), pp. 36-46. (In Russ.) <https://elibrary.ru/eiqepc>

Информация об авторах

Самохвалов Алексей Владимирович, кандидат педагогических наук, доцент, заведующий кафедрой математического моделирования и информационных технологий, Тамбовский государственный университет им. Г.Р. Державина, г. Тамбов, Российская Федерация.

<https://orcid.org/0000-0002-3151-3250>
samohvalov@gmail.com

Соловьев Денис Сергеевич, кандидат технических наук, доцент, доцент кафедры математического моделирования и информационных технологий, Тамбовский государственный университет им. Г.Р. Державина, г. Тамбов, Российская Федерация.

<https://orcid.org/0000-0001-6613-3218>
solovjevdennis@mail.ru

Скворцов Александр Александрович, кандидат педагогических наук, доцент кафедры математического моделирования и информационных технологий, Тамбовский государственный университет им. Г.Р. Державина, г. Тамбов, Российская Федерация.

<https://orcid.org/0000-0003-2041-4000>
skvor_88@mail.ru

Информация о конфликте интересов: авторы заявляют об отсутствии конфликта интересов.

Поступила в редакцию 06.09.2023
Одобрена после рецензирования 20.10.2023
Принята к публикации 25.10.2023

Information about the authors

Alexey V. Samokhvalov, PhD (Education), Associate Professor, Head of Mathematical Modeling and Information Technologies Department, Derzhavin Tambov State University, Tambov, Russian Federation.

<https://orcid.org/0000-0002-3151-3250>
samohvalov@gmail.com

Denis S. Solovjev, PhD (Engineering), Associate Professor, Associate Professor of Mathematical Modeling and Information Technologies Department, Derzhavin Tambov State University, Tambov, Russian Federation.

<https://orcid.org/0000-0001-6613-3218>
solovjevdennis@mail.ru

Alexander A. Skvortsov, PhD (Education), Associate Professor of Mathematical Modeling and Information Technologies Department, Derzhavin Tambov State University, Tambov, Russian Federation.

<https://orcid.org/0000-0003-2041-4000>
skvor_88@mail.ru

Information on the conflict of interests: authors declare no conflict of interests.

Received 06.09.2023
Approved 20.10.2023
Accepted 25.10.2023