

Тренажерно-обучающая система для контроля навыков визуального моделирования программного обеспечения и бизнес-процессов

И.С. Полевщиков^{1,2}, А.П. Рожков², Г.А. Ильин²

¹*Московский государственный университет технологий и управления имени К.Г. Разумовского (ПКУ)*

²*Московский государственный университет пищевых производств*

Аннотация: Рассмотрены особенности математического, алгоритмического и программного обеспечения разрабатываемой тренажерно-обучающей системы (ТОС) для контроля навыков визуального моделирования при подготовке студентов направлений в области автоматизации, информатизации, роботизации. Применение ТОС позволяет автоматически генерировать индивидуальные варианты практических заданий по разработке и анализу визуальных моделей, автоматически оценить правильность выполнения заданий студентами, формировать для студента советующие воздействия (замечания и рекомендации для лучшего понимания изучаемой темы). Применение ТОС в процессе контроля начальных навыков визуального моделирования у студентов позволит: уменьшить необходимость ручного составления большого числа вариантов заданий преподавателем и последующей проверки результатов их выполнения; повысить качество обучения с учетом специфики рассматриваемых направлений подготовки студентов.

Ключевые слова: информационные системы в образовании, тренажерно-обучающая система, автоматизированный контроль знаний и навыков, визуальное моделирование, унифицированный язык моделирования (UML).

1 Введение

При создании автоматизированных систем обработки информации и управления в различных сферах, одной из важных задач является моделирование бизнес-процессов с применением современных графических нотаций (UML, IDEF0, DFD, ARIS и т.д.). Языки визуального моделирования, например, UML, позволяют формализовано описать структуру и поведение сложной программной системы с помощью множества диаграмм [1-3]. Изучение данных языков является одной из составляющих при подготовке специалистов по автоматизации и информатизации [4] (в частности, в сфере пищевых производств).

В настоящее время реализована и частично внедрена в учебный процесс МГУПП и МГУТУ первая версия интеллектуальной информационной системы (ИИС) контроля знаний и навыков при подготовке специалистов по

автоматизации и информатизации (обучающихся, в частности, по направлениям подготовки бакалавров и магистров «Информатика и вычислительная техника», «Программная инженерия», «Управление в технических системах»). Данная версия ИИС позволяет осуществлять дистанционную проверку [4] выполнения заданий (лабораторных, практических, контрольных и других видов работ) преподавателем, что часто представляет весьма трудоемкую деятельность.

Актуальной является задача разработки и внедрения в учебный процесс тренажерно-обучающей системы (ТОС) как одной из подсистем ИИС, реализующей алгоритмы для автоматической генерации и оценки небольших практических заданий по профильным дисциплинам (в частности, в области визуального моделирования) при подготовке данных специалистов (например, при проведении текущего контроля знаний и навыков). Создание данной ТОС основано на развитии существующих работ в области компьютерных тренажерных комплексов и других видов информационных систем, используемых в сфере образования [5-7].

Применение ТОС для контроля навыков визуального моделирования позволит повысить качество учебного процесса при подготовке студентов направлений в области автоматизации и информатизации. Особенности разрабатываемого математического, алгоритмического и программного обеспечения ТОС представлены далее.

2 Математическое и алгоритмическое обеспечение ТОС для контроля навыков визуального моделирования

Наиболее важные функциональные возможности ТОС заключаются в автоматической генерации для каждого студента индивидуальных вариантов небольших практических заданий по визуальному моделированию (в соответствии с настройками преподавателя), автоматическом оценивании

выполнения студентом данных заданий, формировании замечаний и рекомендаций обучаемому. Данные функции основаны на представлении в ТОС знаний об изучаемых объектах профессиональной деятельности (в частности, о специфике визуальных моделей, выполненных в различных графических нотациях).

В процессе разработки практического задания в ТОС преподаватель производит настройку параметров для генерации, выполнения и оценки задания, которые представим множеством:

$$X_{\text{prm.}} = \{X_{\text{obj.}}, X_{\text{tsk.}}, X_{\text{pal.}}, X_{\text{ass.}}\},$$

где $X_{\text{obj.}}$ – множество параметров, определяющих исходный объект (например, математическую модель, алгоритм, программный код) и используемых для генерации индивидуальных вариантов задания;

$X_{\text{tsk.}}$ – множество параметров, определяющих перечень подзадач в составе задания, которые необходимо выполнить обучаемому (например, задачи по построению и анализу объекта);

$X_{\text{pal.}}$ – множество параметров, определяющих свойства палитры для построения объекта обучаемым;

$X_{\text{ass.}}$ – множество параметров, определяющих процесс оценивания выполненного задания.

На рис. 1 в обобщенном виде в форме диаграммы Activity UML показан алгоритм подготовки заданий для студентов с применением ТОС.

Существенной особенностью алгоритма является возможность преподавателя установить необходимые настройки задания (согласно специфике изучаемых видов визуальных моделей), на основе которых будет сформирован индивидуальный вариант каждому студенту.

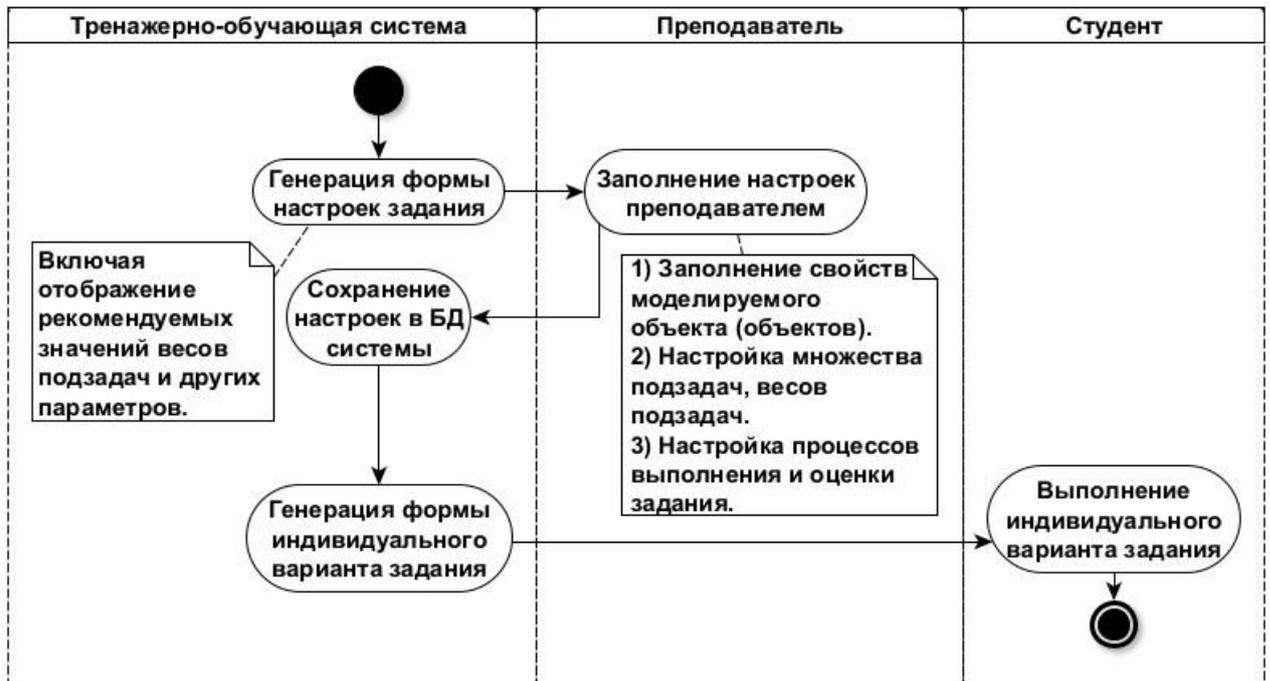


Рис. 1. – Алгоритм подготовки заданий для студентов в ТОС

В практическом аспекте преподаватель производит настройку шаблона индивидуального варианта задания (согласно параметрам множества X_{pm}), заключающуюся в установке значений множества свойств объекта, который необходимо проанализировать или построить обучаемому.

Рассмотрим в качестве примера особенности настройки параметров практического задания по построению или анализу диаграмм Use Case UML. Может быть предусмотрено как непосредственно задание по построению диаграммы на основе текстового описания требований к программному обеспечению (ПО), так и задание по анализу диаграммы (путем ответов на вопросы об элементах данной диаграммы и связях между ними).

Множество параметров, на основе значений которых каждому обучаемому формируется индивидуальный вариант диаграммы Use Case или текстового описания к ней, представим следующим образом:

$X_{\text{obj.}}^{\text{UC}} = \{N_{\text{Act.}}, N_{\text{UC}}, x_A, x_G, x_I, x_E\}$, где $N_{\text{Act.}}$ – число актеров, N_{UC} – число вариантов

использования, x_A – наличие отношения ассоциации, x_G – наличие отношения обобщения, x_I – наличие отношения включения, x_E – наличие отношения расширения.

Потоковый граф, генерируемый в качестве индивидуального варианта задания, определяется предикатом, который представим обобщенно, как $P_{UC}(y_1^{UC}, y_2^{UC}, y_3^{UC}, \dots)$. Переменные данного предиката соответствуют значениям элементов множества $X_{obj.}^{UC}$.

Составим условие на основе параметров множества $X_{obj.}^{UC}$: $(N_{Act.} = 2) \wedge (N_{UC} \in [3; 4]) \wedge (x_A = 1) \wedge (x_G = 1 \vee x_I = 1 \vee x_E = 1)$. Данное высказывание определяет шаблон диаграммы Use Case или ее текстового описания для практического задания в ТОС. Примеры диаграмм Use Case и их текстовых описаний, согласно данным настройкам, представлены в таблице 1.

В зависимости от вида задания – по построению диаграммы или по ее анализу – обучаемому будут соответственно предоставлены индивидуальный вариант текстового описания или самой диаграммы.

На рис. 2 в обобщенном виде в форме диаграммы Activity UML показан алгоритм оценивания выполнения задания на построение объекта (или соответствующей подзадачи в рамках задания) с применением ТОС.

Алгоритм оценивания на рис. 2 основан на автоматическом анализе соответствия структуры математической модели $M_{tr.}$ правильного (эталонного) решения задания и структуры математической модели $M_{st.}$ решения, предложенного студентом, которое может включать в себя ошибки. Модели $M_{tr.}$ и $M_{st.}$ описывают в теоретико-множественном виде структуру и свойства изучаемого объекта (например, диаграммы UML). Математическая модель $M_{tr.}$ эталонного решения задания формируется ТОС автоматически на основе постановки задачи.

Таблица № 1

Примеры диаграмм и их описаний согласно настройкам

№ примера	Диаграмма Use Case UML	Описание диаграммы Use Case UML
1	<p>The diagram shows actor A1 with a hollow triangle arrow pointing to actor A2, indicating that A1 is a generalization of A2. Actor A1 is connected to use case B1 by a solid arrow. Actor A2 is connected to use case B2 by a solid arrow. Use case B1 is connected to use case B3 by a dashed arrow with the label «include».</p>	<ul style="list-style-type: none"> – A1 инициирует выполнение B1. – A2 инициирует выполнение B2. – A2 является потомком A1. – B3 является включаемым по отношению к B1.
2	<p>The diagram shows actor A1 with a hollow triangle arrow pointing to actor A2, indicating that A1 is a generalization of A2. Actor A1 is connected to use case B2 by a solid arrow. Actor A2 is connected to use case B1 by a solid arrow. Use case B2 is connected to use case B3 by a dashed arrow with the label «extend».</p>	<ul style="list-style-type: none"> – A1 инициирует выполнение B2. – A2 инициирует выполнение B1. – A2 является родителем A1. – B3 является расширяющим по отношению к B2.
3	<p>The diagram shows actor A1 connected to use case B1 by a solid arrow. Actor A2 is connected to use case B2 by a solid arrow. Use case B1 is connected to use case B4 by a dashed arrow with the label «extend». Use case B2 is connected to use case B3 by a hollow triangle arrow, indicating that B2 is a generalization of B3. Use case B3 is connected to use case B4 by a hollow triangle arrow, indicating that B3 is a generalization of B4.</p>	<ul style="list-style-type: none"> – A1 инициирует выполнение B1. – A2 инициирует выполнение B2. – B3 является родителем B2 и B4. – B1 является расширяемым по отношению к B4.
4	<p>The diagram shows actor A1 connected to use case B1 by a solid arrow. Actor A2 is connected to use case B4 by a solid arrow. Use case B1 is connected to use case B2 by a dashed arrow with the label «extend». Use case B4 is connected to use case B3 by a dashed arrow with the label «include».</p>	<ul style="list-style-type: none"> – A1 инициирует выполнение B1. – A2 инициирует выполнение B4. – B1 является расширяемым по отношению к B2. – B4 является включающим по отношению к B3.

Допустимы различные частные случаи алгоритма оценивания на рис. 2, учитывающие с определенной степенью точности частично правильное решение задания в зависимости от его специфики. Данные алгоритмы основаны, в частности, на применении нечеткого моделирования [8-10].

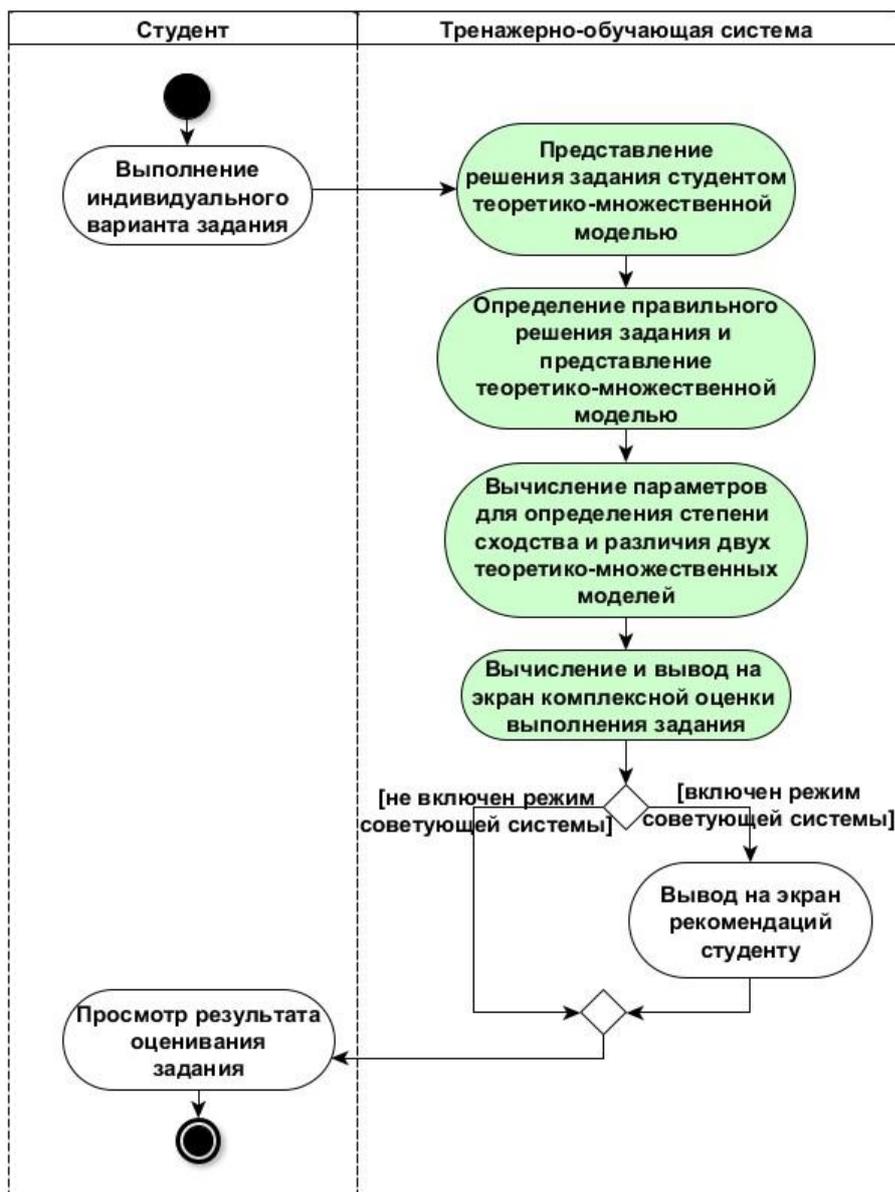


Рис. 2. – Алгоритм оценивания выполнения задания в ТОС

3 Программная реализация ТОС для контроля навыков визуального моделирования

Рассмотрим особенности настройки преподавателем и выполнения

студентом в ТОС практических заданий по моделированию бизнес-процессов и ПО с применением языка UML.

На рис. 3 приведен веб-интерфейс, с помощью которого преподаватель настраивает задание по построению студентом диаграммы Use Case UML на основе текстового описания требований к ПО. Согласно рис. 3, можно выделить следующие основные виды настроек задания:

1) число элементов диаграммы (актеров и вариантов использования), допустимые виды отношений (связей) между элементами;

2) специфика процесса оценивания выполнения задания обучаемым, главным образом, возможность учета частичного совпадения связей между элементами диаграммы студента по сравнению с правильной диаграммой (т.е. преподавателем устанавливается, насколько детализированным будет процесс оценивания);

3) палитра элементов, предоставляемая студенту для построения диаграммы (допустимы палитры элементов разного уровня сложности).

Данные настройки в совокупности позволяют автоматически сформировать каждому обучаемому индивидуальный вариант задания, определяют особенности оценивания и выполнения задания в системе. Веб-интерфейс на рис. 3 может быть расширен путем дополнительных настроек (например, допустимое время выполнения задания).

На рис. 4 показан интерфейс выполнения задания по построению диаграммы Use Case UML студентом. Студенту предоставляется описание функциональных требований к программе, сформированное автоматически в соответствии с настройками преподавателя (рис. 3).

С помощью мини-редактора обучаемый строит диаграмму, выбирая названия актеров и вариантов использования из списка допустимых. На рис. 4 видно, что студентом допущена ошибка при указании отношений между вариантами использования В1 и В4, В3 и В4.

Формулировка задания:

Построить диаграмму Use Case UML на основе следующего описания требований к программе:

Настройки задания:

Настройка шаблона	Число актеров: от <input type="text" value="2"/> до <input type="text" value="2"/> . Число вариантов использования: от <input type="text" value="3"/> до <input type="text" value="4"/> . Виды отношений (помимо ассоциации): <input checked="" type="checkbox"/> включение <input checked="" type="checkbox"/> расширение <input checked="" type="checkbox"/> обобщение Число видов отношений (помимо ассоциации): от <input type="text" value="2"/> до <input type="text" value="2"/> .								
Настройка процесса оценки	<input type="radio"/> учитывать только полное совпадение связей <input checked="" type="radio"/> учитывать частичное совпадение связей: <table border="1"><thead><tr><th>Критерий соответствия эталонной диаграммы и диаграммы обучаемого</th><th>Вес</th></tr></thead><tbody><tr><td>наличие связи между элементами</td><td><input type="text" value="1"/></td></tr><tr><td>одинаковое направление связи</td><td><input type="text" value="2"/></td></tr><tr><td>одинаковый вид связи</td><td><input type="text" value="2"/></td></tr></tbody></table>	Критерий соответствия эталонной диаграммы и диаграммы обучаемого	Вес	наличие связи между элементами	<input type="text" value="1"/>	одинаковое направление связи	<input type="text" value="2"/>	одинаковый вид связи	<input type="text" value="2"/>
Критерий соответствия эталонной диаграммы и диаграммы обучаемого	Вес								
наличие связи между элементами	<input type="text" value="1"/>								
одинаковое направление связи	<input type="text" value="2"/>								
одинаковый вид связи	<input type="text" value="2"/>								
Палитра элементов	<input type="radio"/> упрощенная (элементы и связи в требуемом количестве представлены в палитре) <input checked="" type="radio"/> стандартная (обучаемый создает нужное число элементов и связей) Перейти в редактор палитры								

Рис. 3. – Интерфейс настройки задания по построению диаграммы Use Case

При автоматическом оценивании задания (после нажатия ссылки «Проверить») данные ошибки будут учитываться, согласно алгоритму на рис. 2. При включенном режиме советующей системы студенту также предоставляются рекомендации по допущенным ошибкам.

В примере выше приведена упрощенная диаграмма UML, без связи с какой-либо предметной областью. Актеры и варианты использования обозначаются с помощью определенных комбинаций букв и цифр, например, A1, A2, A3, ...AN, B1, B2, B3..., BM, где N и M – общее число актеров и вариантов использования на диаграмме соответственно.

Выбор элементов	Область для построения диаграммы Use Case UML	Свойства элемента
		<p>Название элемента:</p> <ul style="list-style-type: none"> <input type="radio"/> A1 <input type="radio"/> A2 <input type="radio"/> B1 <input type="radio"/> B2 <input type="radio"/> B3 <input checked="" type="radio"/> B4

[Проверить](#)

Рис. 4. – Веб-интерфейс выполнения задания студентом

4 Заключение

1) Разрабатываемая ТОС для контроля навыков визуального моделирования позволяет автоматически (в соответствии с настройками преподавателя), генерировать индивидуальные варианты практических заданий по разработке и анализу визуальных моделей, автоматически оценивать правильность выполнения заданий студентами, формировать для студента советуемые воздействия (замечания и рекомендации для лучшего понимания изучаемой темы).

2) Применение ТОС в процессе контроля начальных навыков обучаемых в области автоматизации, информатизации, роботизации, позволит:

– уменьшить необходимость ручного составления большого числа вариантов заданий преподавателем и последующей проверки результатов их выполнения, что, как правило, отнимает много времени и у преподавателя, и

у студента (ожидание результатов проверки);

– повысить качество обучения с учетом специфики рассматриваемых направлений подготовки студентов.

3) В практическом аспекте, ТОС можно использовать при проведении контрольных работ в ходе аудиторных занятий, при выполнении лабораторных и практических работ (например, одна из задач в рамках работы заключается в построении студентом визуальной модели с возможностью дальнейшей автоматической проверки), для защиты отчетов по лабораторным и практическим работам, при выполнении заданий в рамках внутрифирменного обучения сотрудников предприятий.

Литература

1. Картечина Н.В., Трейгер В.В., Дорохова А.М., Шацкий В.А., Гущина А.А., Чиркин С.О. Моделирование в UML автоматизированной подсистемы инвентаризации предприятия // Наука и Образование. 2021. Т. 4. № 3. URL: opusmgau.ru/index.php/see/article/view/3842/3824.

2. Макеева О.В., Сартаков М.В., Чернов Е.А. Моделирование информационных процессов с помощью UML // Инновации и инвестиции. 2021. № 9. С. 121-125.

3. Муллахметов А.Р. Проектирование автоматизированной информационной системы с использованием UML-технологии в образовательных организациях // Вестник Набережночелнинского государственного педагогического университета. 2022. № 1 (36). С. 59-61.

4. Полевщиков И.С. Методика разработки практических задач для автоматизированного контроля знаний и навыков при обучении ИТ-специалистов // Инженерный вестник Дона. 2020. №12. URL: ivdon.ru/ru/magazine/archive/n12y2020/6785.

5. Файзрахманов Р.А., Шкляев Ф.И., Шушарина А.В. Концепция

информационной системы моделирования компьютерных тренажеров для формирования сенсомоторных навыков // Информационно-измерительные и управляющие системы. 2019. Т. 17. № 4. С. 28-37.

6. Зорин Ю.А. Использование алгоритмов комбинаторной генерации при построении генераторов тестовых заданий // Дистанционное и виртуальное обучение. 2013. № 6 (72). С. 54-59.

7. Aldwairi M. Evaluating virtual laboratory platforms for supporting on-line information security courses // Global Journal of Engineering Education. 2022. V. 24. №2. pp. 143-148.

8. Файзрахманов Р.А., Полевщиков И.С. Автоматизированное управление формированием профессиональных навыков оператора роботизированной системы с использованием нечеткой логики // Инженерный вестник Дона. 2015. №4. URL: ivdon.ru/ru/magazine/archive/n4y2015/3283.

9. Ismagilov I.I., Alsaied G. Fuzzy regression analysis using trapezoidal fuzzy numbers // Industrial Engineering and Management Systems. 2020. V. 19. № 4. pp. 896-900.

10. Катасёв А.С. Нейронечеткая модель формирования нечетких правил для оценки состояния объектов в условиях неопределенности // Компьютерные исследования и моделирование. 2019. Т. 11. № 3. С. 477-492.

References

1. Kartechina N.V., Treyger V.V., Dorokhova A.M., Shatskiy V.A., Gushchina A.A., Chirkin S.O. Nauka i Obrazovanie. 2021. V. 4. № 3. URL: opusmgau.ru/index.php/see/article/view/3842/3824.

2. Makeeva O.V., Sartakov M.V., Chernov E.A. Innovatsii i investitsii. 2021. № 9. pp. 121-125.

3. Mullakhmetov A.R. Vestnik Naberezhnochelninskogo gosudarstvennogo



pedagogicheskogo universiteta. 2022. № 1 (36). pp. 59-61.

4. Polevshchikov I.S. Inzhenernyj vestnik Dona, 2020, №12. URL: ivdon.ru/ru/magazine/archive/n12y2020/6785.

5. Fayzrakhmanov R.A., Shklyayev F.I., Shusharina A.V. Informatsionno-izmeritel'nye i upravlyayushchie sistemy. 2019. V. 17. № 4. pp. 28-37.

6. Zorin Yu.A. Distantcionnoe i virtual'noe obuchenie. 2013. № 6 (72). pp. 54-59.

7. Aldwairi M. Global Journal of Engineering Education. 2022. V. 24. №2. pp. 143-148.

8. Fayzrakhmanov R.A., Polevshchikov I.S. Inzhenernyj vestnik Dona, 2015, №4. URL: ivdon.ru/ru/magazine/archive/n4y2015/3283.

9. Ismagilov I.I., Alsaied G. Industrial Engineering and Management Systems. 2020. V. 19. № 4. pp. 896-900.

10. Katasev A.S. Komp'yuternye issledovaniya i modelirovanie. 2019. V. 11. № 3. pp. 477-492.