

УДК 004.89:378.147

DOI 10.62595/1819-5245-2025-2-99-109

ГЕНЕРАЦИЯ И ВЕРИФИКАЦИЯ ЗАДАНИЙ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ С ВАРИАНТАМИ ПО УЧЕБНЫМ ДИСЦИПЛИНАМ ИТ-ЦИКЛА С ИСПОЛЬЗОВАНИЕМ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

О. Д. АСЕНЧИК, А. А. ЯСТРЕБОВ, В. Н. ИВАШКО

*Учреждение образования «Гомельский государственный
технический университет имени П. О. Сухого»,
Республика Беларусь*

Предложен алгоритм и прототип программного комплекса на базе больших языковых моделей для автоматизации процесса создания и верификации уникальных заданий для лабораторных работ по дисциплинам сферы информационных технологий. Основу решения составляет конвейер из трех агентов, обеспечивающих создание эталонного мастер-задания, параметрическое варьирование уникальных вариантов, контроль их сложности и автоматическое исправление ошибок. Проведена оценка эффективности и корректности работы решения. Результаты тестирования подтвердили высокое качество генерируемых заданий для лабораторных работ и соблюдение критериев уникальности вариантов заданий. Отмечено, что решение позволяет существенно сократить временные затраты преподавателей, сохраняя при этом педагогический контроль над содержанием учебного процесса.

Ключевые слова: большие языковые модели, LLM, генерация учебных заданий, искусственный интеллект, образование.

Для цитирования. Асенчик, О. Д. Генерация и верификация заданий для лабораторных работ с вариантами по учебным дисциплинам ИТ-цикла с использованием больших языковых моделей / О. Д. Асенчик, А. А. Ястребов, В. Н. Ивашко // Вестник Гомельского государственного технического университета имени П. О. Сухого. – 2026. – № 2 (105). – С. 99–109. <https://doi.org/10.62595/1819-5245-2026-2-99-109>

GENERATION AND VERIFICATION OF ASSIGNMENTS FOR LABORATORY WORK WITH OPTIONS FOR IT-CYCLE ACADEMIC DISCIPLINES USING LARGE LANGUAGE MODELS

O. D. ASECHNIK, A. A. YASTREBOV, V. N. IVASHKO

*Sukhoi State Technical University of Gomel,
the Republic of Belarus*

An algorithm and a prototype software system based on large language models are proposed to automate the creation and verification of unique assignments for laboratory work in information technology disciplines. The solution is based on a three-agent pipeline that provides the generation of a reference master assignment, parametric variation of unique task instances, complexity control, and automatic error correction. An evaluation of the efficiency and correctness of the proposed solution was carried out. The testing results confirmed the high quality of the generated laboratory assignments and compliance with the requirements for the uniqueness of task variants. It was noted that the proposed solution makes it possible to significantly reduce instructors' time expenditure while preserving pedagogical control over the content of the educational process.

Keywords: large language models, LLM, generation of educational tasks, artificial intelligence, education.

For citation. Asenichik O. D., Yastrebov A. A., Ivashko V. N. Generation and verification of assignments for laboratory work with options for IT-cycle academic disciplines using large language models. *Vestnik Gomel'skogo gosudarstvennogo tekhnicheskogo universiteta imeni P. O. Sukhogo*, 2026, no. 2 (105), pp. 99–109 (in Russian). <https://doi.org/10.62595/1819-5245-2026-2-99-109>

Введение

Одной из актуальных проблем в сфере образования является обеспечение самостоятельности выполнения студентами учебных заданий. В дисциплинах, изучаемых будущими IT-специалистами (дисциплины IT-цикла), она выражается, прежде всего, в необходимости обеспечить каждого студента индивидуальным вариантом лабораторного задания, что необходимо для объективной оценки практических навыков и исключения возможности заимствования чужих решений. Подготовка нескольких десятков уникальных вариантов задания для каждой из лабораторных работ по каждой из преподаваемых дисциплин представляет собой очень трудоемкую задачу, требующую от преподавателя значительных временных затрат.

Появление и развитие больших языковых моделей (LLM) открывает новые возможности для автоматизации процессов подготовки учебных материалов [1]. Эти модели, обученные на масштабных корпусах самых разных текстов, включая техническую документацию, учебные пособия, образцы программного кода, демонстрируют способность генерировать осмысленные, структурированные и адаптируемые к текущему содержанию запроса тексты. Подход, предполагающий генерацию содержания всех лабораторных работ и их вариантов в рамках одного запроса к LLM посредством прямого обращения к языковым моделям через предоставляемый их производителями интерфейс, оказался недостаточно управляемым и эффективным из-за ограничений размера контекстного окна моделей, большой единовременной нагрузки на ресурсы серверов с установленной LLM, неустойчивости сетевого соединения между сервером и клиентом. Полученные результаты, как правило, требуют существенной доработки преподавателем и других уточняющих запросов к LLM для обеспечения семантической уникальности получаемых вариантов, их корректности и соответствия учебным целям дисциплины. Устранение перечисленных проблем требует разработки специального алгоритма многоступенчатой генерации и механизма верификации получаемых текстов лабораторных работ.

В научной литературе задача генерации учебных заданий активно исследуется с позиций применения архитектуры трансформеров [1, 2] и конструирования запросов к языковым моделям [3]. В работе [4] описан метод генерации структурно идентичных (изоморфных), но с разными данными математических задач с использованием LLM: сначала формулируется общая структура задачи (например, тип уравнения, шаги решения), а затем LLM подбирает конкретные параметры (числа, переменные, сюжет). Такой подход является перспективным для обеспечения стабильной управляемой генерации большого количества вариантов одинаковой сложности, снижения возможных галлюцинаций. В настоящей работе мы распространяем идею генерации на основе шаблона на случай генерации вариантов заданий для лабораторных работ в менее детерминированной области – для дисциплин IT-цикла.

С другой стороны, описаны решения на основе взаимодействующих программных единиц, использующих возможности LLM (LLM-агентов), где агенты применяются для взаимной проверки и верификации сгенерированных заданий [5]. Автоматическая оценка таких заданий с привлечением LLM в роли независимого эксперта [6] продемонстрировала свою состоятельность.

Целью настоящей работы является разработка и экспериментальная оценка работы алгоритма и соответствующего программного комплекса для генерации и верификации уникальных вариантов заданий для лабораторных работ по дисциплинам IT-цикла на базе больших языковых моделей.

Алгоритм параметрической генерации вариантов заданий

Предлагаемая формализованная постановка задачи может быть представлена следующим образом. Результат процесса создания учебного задания P описывается в виде кортежа, состоящего из четырех элементов:

$$P = (Q, M, N, A), \quad (1)$$

где Q – текст названия и краткого описания работы; $M = M(Q)$ – текст шаблона задания (мастер-задание), который генерирует LLM, и которое верифицируется преподавателем; N – набор варьируемых параметров; A – итоговый результат – вариант задания.

Требуется создание набора из k задач $\{P_1, P_2, \dots, P_k\}$ на основе набора (M, Q, N_k) , при этом каждая сгенерированная задача должна быть уникальна и иметь примерно одинаковый уровень сложности. При этом матрица N_k состоит из k конкретных значений наборов параметров варьирования из N , сгенерированных LLM.

В основу алгоритмов программного комплекса положена предложенная авторами классификация способов варьирования заданий лабораторных работ [7]. Ее суть заключается в первичном составлении шаблонного задания (мастер-задания), к которому затем подготавливается серия индивидуальных вариантов за счет варьирования значений одного или нескольких параметров. Предложен следующий набор: предметная область: адаптация единой структуры задания под различные прикладные сценарии и контексты; входные данные: предоставление индивидуальных наборов данных для решения идентичной задачи; формат выходных данных: способы представления результатов (типы файлов, форматы отчетов и графиков); алгоритмические требования: предписание конкретных методов реализации, структур данных или алгоритмических подходов; ограничения на ресурсы и модули: лимиты на использование памяти и времени, а также требования к программным библиотекам; стек инструментов и технологий: выполнение задания с использованием различных языков программирования, фреймворков или сред разработки.

В предлагаемом решении мастер-задание первично генерируется языковой моделью исходя из краткого описания лабораторной работы и текста описывающего запроса (промпта). У преподавателя есть возможность внести правки в сгенерированное мастер-задание перед его утверждением. Активное участие преподавателя позволяет получить адекватный учебный материал с учетом его авторского взгляда.

После утверждения мастер-задания задается набор параметров варьирования с возможностью сохранения значений параметров для их переноса в последующие лабораторные работы, обеспечивая их тематическую связность.

Генерацию вариантов заданий к лабораторной работе выполняет разработанный многоступенчатый конвейер, который является ключевым элементом новизны программного комплекса. Он включает в себя механизмы обратной связи для обеспечения уникальности генерируемых вариантов, проверки их соответствия тематике задания, а также для обеспечения одинакового уровня сложности всех вариантов.

В совокупности предложенные алгоритмические решения образуют законченный цикл автоматизированной генерации и верификации вариантов заданий для лабораторных работ по дисциплинам ИТ-цикла.

Разработанный алгоритм параметрической многоступенчатой генерации вариантов заданий для лабораторной работы реализован как трехэтапный конвейер, общая схема которого представлена на рис. 1.

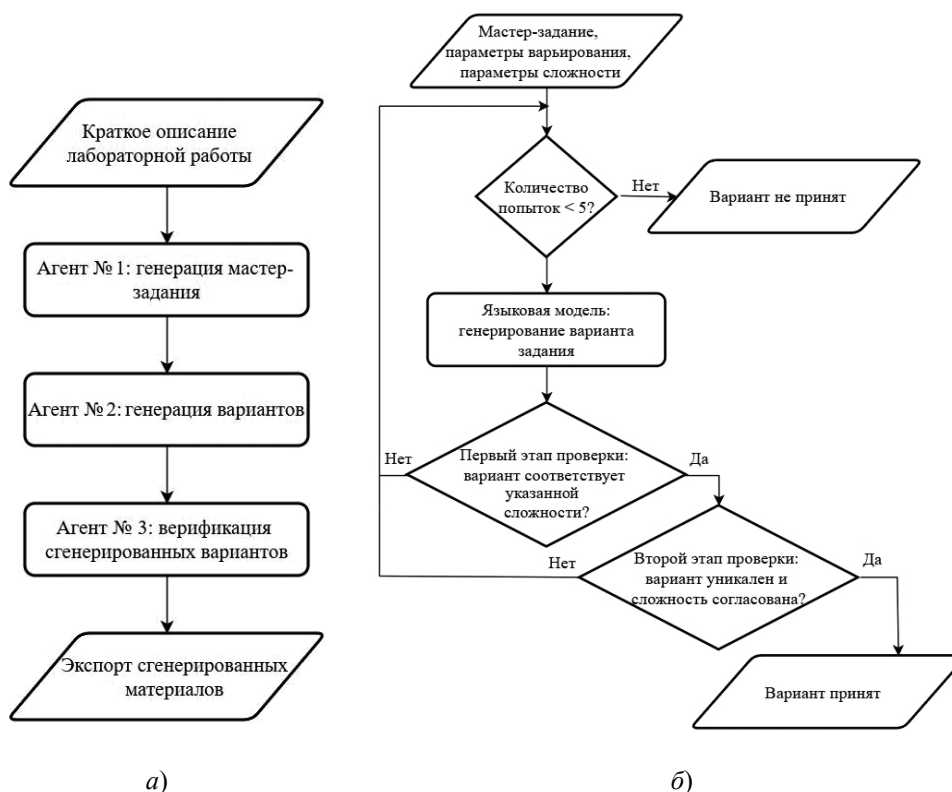


Рис. 1. Схема алгоритма генерации вариантов заданий:
 а – общий вариант; б – детализированная схема алгоритма работы агента № 2

Каждая ступень (агент) для выполнения своей задачи использует LLM, динамически формируя для нее текст запроса (промпт). В тексте запроса описывается задача, данные к ней, а также задается формат ответа для дальнейшей его обработки агентом.

Агент № 1 – сервис, формирующий мастер-задание по краткому описанию лабораторной работы. При этом описывающий требования к его структуре промпт может настраиваться; нами был использован следующий: «Требования: четко сформулированная цель; постановка задачи; спецификация входных и выходных данных; список требований (что именно необходимо реализовать, ограничения, граничные случаи)». LLM генерирует текст мастер-задания согласно описанию и промпту. Этот текст преподаватель может отредактировать и должен утвердить.

Агент № 2 – сервис, формирующий уникальные варианты одинакового уровня сложности согласно заданным параметрам варьирования и уровню сложности лабораторной работы. Для каждого из заданного числа вариантов выполняется итерационный цикл с настраиваемым количеством повторных попыток. На каждой итерации LLM формирует вариант-кандидат задания с учетом следующих параметров: перечень выбранных способов варьирования и указанных для них параметров, а также целевых параметров сложности варианта задания, представленных в виде оценочного минимального и максимального количества часов, необходимых для выполнения задания.

Этот сервис также обеспечивает механизм сквозного переноса значения варьируемого параметра между лабораторными работами. Если для какого-либо способа варьирования включен режим сохранения значений между лабораторными работами, сервис учитывает значения параметров варьирования, назначенные соответствующему варианту в предшествующих работах дисциплины. Например, если указан

перенос значений для способа варьирования по предметной области, то для каждого генерируемого варианта будет выполнен перенос предметной области из соответствующего ему варианта предыдущей лабораторной работы.

Вариант-кандидат проходит двухуровневую проверку соответствия заданному уровню сложности. При генерации варианта задания языковая модель выполняет оценку его сложности и помещает эти оценки в ответ с текстом варианта задания. На первом уровне выполняется проверка соответствия этих оценок целевым параметрам сложности варианта задания (без обращения к языковой модели). На втором уровне такую же оценку выполняет LLM: соответствует ли фактическая сложность варианта задания ожидаемому уровню.

Несоответствие на первом уровне и на втором уровне на первой итерации цикла генерации варианта задания приводит к повторной генерации с передачей обоснования несоответствия в следующий запрос к LLM. В последующих итерациях при несоответствии на втором уровне проверки повторная генерация не выполняется (сгенерированный вариант задания считается удовлетворительным).

Похожий алгоритм выполняется и для проверки сгенерированного варианта задания на семантическую уникальность. Эта проверка выполняется путем запроса к языковой модели. Если языковая модель считает вариант неуникальным, то в ответе она описывает обоснование, а сгенерированный вариант отклоняется. Далее выполняется повторная генерация варианта задания с учетом обоснования отклоненного варианта.

Принятый вариант задания сохраняется вместе с вычисленным для него криптографическим идентификатором содержимого для исключения его точных дубликатов.

Агент № 3 – сервис, выполняющий верификацию качества сгенерированных вариантов. LLM оценивает соответствие варианта мастер-заданию и формирует структурированный отчет с перечнем замечаний. При неудовлетворительной оценке та же LLM в роли редактора исправляет вариант согласно перечню замечаний, после чего оценка производится повторно.

Архитектура программного комплекса

Программный комплекс представляет собой веб-приложение с клиент-серверной архитектурой [8]. Серверная часть построена на платформе .NET 10.0 и реализует четырехслойную архитектуру: доменный слой, содержащий сущности предметной области; слой приложения, определяющий абстракции сервисов; инфраструктурный слой, реализующий логику генерации, механизмы взаимодействия с LLM и хранение данных в локальной базе данных; слой веб-интерфейса, предоставляющий программный интерфейс типа REST API.

Для обеспечения воспроизводимости результатов и гибкости вычислений в системе реализована поддержка различных провайдеров LLM: локальных решений через Ollama-совместимые API, а также облачных моделей через агрегатор OpenRouter. Задачи генерации и верификации выполняются асинхронным фоновым сервисом, что позволяет не блокировать пользовательский интерфейс.

Дополнительно в инфраструктурном слое реализован компонент, предназначенный для автоматического формирования вспомогательных материалов: теоретического раздела и перечня контрольных вопросов к лабораторной работе.

Генерация вспомогательных материалов осуществляется посредством отдельного вызова LLM. Для оптимизации использования вычислительных ресурсов и предотвращения избыточных запросов реализован механизм кеширования ответов на основе хэш-функции SHA-256. Сгенерированные теоретические материалы и вопросы проходят процедуру верификации и автоматически включаются в состав итогового документа при экспорте в формат Microsoft Word (.docx).

Клиентская веб-часть предоставляет преподавателю рабочую среду с разделением по функциональным вкладкам: редактирование и утверждение мастер-задания, настройка методов варьирования, управление вариантами заданий, просмотр отчетов верификации и выгрузка итоговых документов в формат Microsoft Word (.docx).

Методика и результаты автоматизированного и экспертного тестирования

Для проверки эффективности разработанного программного комплекса и предложенной методики разработаны два программных эмулятора, имитирующих деятельность преподавателя и оценивающих отклик системы.

Первый эмулятор предназначен для проверки способности различных LLM генерировать значительный объем уникальных вариантов при варьировании заданных параметров. Второй эмулятор оценивал качественные характеристики: семантическую уникальность заданий, логическую связность и уникальность вариантов заданий.

Процесс испытаний в обоих случаях следовал единому алгоритму:

1. Входные данные – план испытаний в табличном формате, который содержит сценарии, определяющие дисциплину, количество работ и параметры варьирования.
2. Исполнение сценария, т. е. последовательное создание структуры дисциплины, генерация и утверждение мастер-заданий (где роль рецензента выполняет отдельный агент на базе LLM), настройка профилей варьирования и запуск генерации вариантов.
3. Формирование структурированного журнала с результатами генерации, который обрабатывается аналитическим модулем с привлечением LLM в роли эксперта-судьи.

Оценка результатов проводилась по девяти формализованным критериям, разделенным на две группы: бинарные (оценка структурной целостности) и качественные (оценка содержания по шкале от 1 до 5) (табл. 1).

Таблица 1

Критерии оценки сгенерированных заданий для лабораторных работ

Группа критериев	Критерий	Описание	Шкала
Бинарные	Соответствие дисциплине	Тематика заданий отвечает содержанию дисциплины	Да/нет
	Различие лабораторных работ	Задания по разным работам отличаются по тематике и задачам	Да/нет
	Логичность последовательности	Порядок работ соответствует логике изучения дисциплины	Да/нет
	Различие вариантов	Варианты одной работы отличаются предметной областью и алгоритмами	Да/нет
	Единообразие сложности	Варианты одной работы имеют сопоставимый уровень трудоемкости	Да/нет
Качественные	Корректность	Постановка задачи корректна и однозначна	1–5
	Качество	Изложение соответствует профессиональному стандарту	1–5
	Полнота	Задание содержит все необходимые разделы и требования	1–5
	Ясность	Формулировки понятны целевой аудитории	1–5

Для проверки возможностей различных LLM работать с алгоритмами программного комплекса были выбраны модели разного размера, как правило, с открытым исходным кодом и доступные путем вызова через API сервиса ollama.com в рамках бесплатной квоты доступа: gemma3:4b, gemma3:12b, gemma3:27b, gemma4:31b, gpt-oss:20b, gpt-oss:120b, deepseek-v3.2, qwen 3.5, cogito-2.1:671b, gemini 3.1 flash preview. Суть тестирования заключается в проверке возможности каждой LLM генерировать 15 уникальных вариантов по одной лабораторной работе по дисциплине «Объектно-ориентированное проектирование и программирование» при варьировании двух параметров: предметной области и алгоритмических требований. Количество повторных генераций в случае проявления повторов ограничивалось пятью попытками. Результаты генерации вариантов оценивались с использованием LLM deepseek-v3.2 (кроме результатов работы самой deepseek-v3.2; в этом случае в качестве судьи была использована модель cogito-2.1). В табл. 2 приведены результаты работы этого эмулятора.

Таблица 2

Возможности различных LLM по генерации уникальных вариантов

Модель	Доля сгенерированных вариантов, %	Количество повторов	
		верификации	генерации
gemma3:4b	7	–	–
gemma3:12b	7	–	–
gemma3:27b	7	–	–
gemma4:31b	47	–	–
gpt-oss:20b	100	1	0
gpt-oss:120b	100	0	0
deepseek-v3.2	100	0	0
qwen 3.5	100	1	1
cogito-2.1:671b	100	0	0
gemini 3.1 flash preview	100	1	1

Повторы верификации – это дополнительные циклы повторной верификации и автоматического исправления вариантов, не прошедших первичную оценку. Повторы генерации – дополнительные циклы генерации новых вариантов взамен тех, которые не удалось исправить.

Для моделей gemma3/gemma4 генерация вариантов не завершилась успешно (доля сгенерированных вариантов составляет 7 и 47 % соответственно), до стадии верификации процесс не дошел, поэтому в соответствующих ячейках стоят прочерки. На основании полученных результатов для проведения дальнейших расширенных серий испытаний программного комплекса была выбрана модель cogito-2.1:671b, так как она продемонстрировала успешность генерации и точность следования инструкциям промпта без обращений к механизмам повторной верификации и генерации. Для использования подойдут также модели gpt-oss:20b, gpt-oss:120b и deepseek-v3.2, так как они тоже показали хорошие результаты.

Для проверки качества генерации и работы ключевого механизма варьирования проведено тестирование в четырех сериях испытаний, охвативших в совокупности 16 сценариев по двум дисциплинам ИТ-цикла: «Компьютерные сети» и «Объектно-ориентированное проектирование и программирование». Конфигурации сценариев систематически варьировали следующие параметры: для «Компьютерных сетей»

менялись предметная область и стек технологий; для «Объектно-ориентированного проектирования и программирования» – предметная область и алгоритмические требования.

Количество лабораторных работ задавалось в зависимости от количества варьируемых параметров: если варьировался один параметр, то система генерировала три лабораторных работы; если два параметра, то шесть работ.

Для каждой отдельной лабораторной работы система генерировала шесть вариантов заданий вне зависимости от количества варьируемых параметров.

Из-за большого объема тестовых данных оценка сгенерированных работ согласно описанным ранее критериям выполнялась автоматизированно: алгоритмически и с использованием LLM deepseek-v3.2.

По итогам оценки 72-х лабораторных работ по всем сценариям получены следующие средние числовые оценки качества сгенерированных заданий (табл. 3).

Таблица 3

Средние числовые оценки качества сгенерированных заданий

Название дисциплины	Качественные критерии оценки				
	Ясность	Полнота	Качество	Корректность	Общая оценка
Компьютерные сети	4,86	4,64	4,53	4,78	4,70
Объектно-ориентированное проектирование и программирование	4,97	4,89	4,89	4,94	4,92

Анализ оценок свидетельствует об относительно высоком качестве полученных заданий. Результаты для дисциплины «Объектно-ориентированное проектирование и программирование» оказались несколько выше (общая оценка – 4,92), чем для «Компьютерных сетей» (4,70). Общая оценка – это среднее арифметическое четырех критериев. Различие можно объяснить спецификой применяемых методов варьирования: алгоритмические требования для заданий по программированию легче формализуются и обрабатываются LLM, в то время как варьирование стека сетевых технологий требует от модели более глубокого контекстного согласования, что в единичных случаях приводило к незначительному снижению баллов по критерию «Качество» (4,53).

Схожие результаты были получены и при проверке по бинарным критериям качества. Качество сгенерированных материалов оценивалось агрегированно по каждой дисциплине. Все итоговые результаты получили положительную бинарную оценку («Да»). Стоит отметить, что такие показатели достигнуты благодаря заложенному в алгоритм генерации итерационному подходу: если вариант не проходил проверку, программный комплекс автоматически выполнял повторную генерацию варианта задания.

В дополнение к автоматизированному тестированию была проведена серия ручных (экспертных) испытаний программного комплекса, в ходе которого преподаватель непосредственно, используя графический интерфейс, анализировал качество генерируемых мастер-заданий, выполнял их корректировку и оценивал качество сгенерированных вариантов к мастер-заданиям. Тестирование проведено для дисциплин «Разработка мобильных Android-приложений», «Современные технологии серверной разработки» и «Распределенные информационные системы». Конечная цель тестирования – проверка качества генерации серий лабораторных работ в рамках учебной программы дисциплины.

Например, для дисциплины «Разработка мобильных Android-приложений» сгенерировано четыре лабораторные работы. Для каждой лабораторной работы сгенерировано 15 вариантов. Темы лабораторных работ и способы варьирования следующие:

- «Основы Jetpack Compose и обработка событий» – варьирование по предметной области и формату выходных данных.
- «Сложные компоненты UI, навигация и архитектура ИС» – варьирование по предметной области, алгоритмическим требованиям и формату выходных данных.
- «Работа с локальной базой данных Room» – варьирование по предметной области (с переносом из предыдущей работы).
- «Взаимодействие с внешними API через Retrofit» – варьирование по предметной области (с переносом из предыдущей работы) и формату выходных данных.

Для дисциплины «Современные технологии серверной разработки» сгенерировано четыре работы, а для «Распределенные информационные системы» – шесть работ.

В общей сумме сгенерировано 14 лабораторных работ. В ходе испытаний было замечено, что даже незначительная корректировка мастер-задания преподавателем на начальном этапе (например, уточнение формулировок или ограничений) приводит к повышению качества генерируемых вариантов.

Программный комплекс успешно сгенерировал комплект лабораторных работ в рамках учебных программ дисциплин. При этом следует отметить, что сгенерированные варианты заданий имеют семантическую уникальность; реализованный механизм сквозного переноса предметной области отработал корректно, заданный контекст варианта сохранялся на протяжении нескольких лабораторных работ.

Также программный комплекс успешно продемонстрировал способность сочетать сразу несколько способов варьирования. Например, алгоритм позволял одновременно комбинировать предметную область и формат выходных данных (например, вывод данных в виде таблицы, карточек, сохранения в файл или в базу данных и т. д.), не нарушая при этом общей структуры и реализуемости задания.

Заключение

Разработанный на базе платформы .NET программный комплекс реализует алгоритм работы трехэтапного конвейера, который обеспечивает полный цикл подготовки лабораторных работ: от генерации мастер-задания до экспорта готовых уникальных вариантов в формате Microsoft Word (.docx).

Результаты автоматизированного и экспертного тестирования подтвердили высокую эффективность предложенных алгоритмов. Автоматизированное тестирование на выборке из 72-х лабораторных работ показало, что система генерирует задания с высокой степенью ясности и корректности, при этом сгенерированные варианты заданий удовлетворяют критериям уникальности, тематического соответствия и единообразия сложности. Успешная генерация 14-ти комплексных лабораторных работ в ходе ручного тестирования доказала работоспособность механизма сквозного переноса контекста предметной области.

Использование программного комплекса позволит снизить нагрузку на профессорско-преподавательский состав при подготовке к занятиям и обеспечит необходимые условия для самостоятельности выполнения учебных заданий студентами. При этом педагогический контроль над содержанием учебного материала полностью сохраняется за счет обязательного этапа утверждения и редактирования мастер-заданий преподавателем.

Литература

1. Асенчик, О. Д. Использование больших языковых моделей для создания учебных материалов по дисциплине «Базы данных» / О. Д. Асенчик // Цифровая трансформация. – 2025. – Т. 31, № 4. – С. 5–14. – DOI 10.35596/1729-7648-2025-31-4-5-14
2. Language Models are Few-Shot Learners / T. Brown, B. Mann, N. Ryder [et al.] // *Advances in Neural Information Processing Systems*. – 2020. – Vol. 33. – P. 1877–1901.
3. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models / J. Wei, X. Wang, D. Schuurmans [et al.] // *Advances in Neural Information Processing Systems*. – 2022. – Vol. 35. – P. 24824–24837.
4. Kim, J.-H. Computational Blueprints: Generating Isomorphic Mathematics Problems with Large Language Models / J.-H. Kim, J. Nam, G. Jo // *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Suzhou, China, 2025* / Association for Computational Linguistics. – Suzhou, 2025. – P. 1389–1401.
5. MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework / S. Hong, X. Zheng, J. Chen [et al.] // *Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 2024* / OpenReview. – Vienna, 2024. – P. 1–26.
6. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena / L. Zheng, W.-L. Chiang, Y. Sheng [et al.] // *Advances in Neural Information Processing Systems, Red Hook, NY, 2023* / Curran Associates, Inc. – Red Hook, 2023. – Vol. 36. – P. 46561–46585.
7. Асенчик, О. Д. Классификация способов варьирования заданий лабораторных работ по учебным дисциплинам сферы информационных технологий / О. Д. Асенчик, А. А. Ястребов // Проблемы современного образования в техническом вузе : материалы IX Междунар. науч.-метод. конф., Гомель, 23–24 окт. 2025 г. / М-во образования Респ. Беларусь, Гомел. гос. техн. ун-т им. П. О. Сухого. – Гомель, 2025. – С. 62–64.
8. LabGenerator: GitHub repository. – URL: <https://github.com/Olgasn/LabGenerator/tree/main> (date of access: 14.04.2026).

References

1. Asenchik O. D. Using large language models to create teaching materials for the discipline “Databases”. *Cifrovaja transformacija = Digital Transformation*, 2025, vol. 31, no. 4, pp. 5–14 (in Russian). <https://doi.org/10.35596/1729-7648-2025-31-4-5-14>
2. Brown T., Mann B., Ryder N., Subbiah M., Kaplan J., Dhariwal P., Neelakantan A., Shyam P., Sastry G., Askell A., Agarwal S., Herbert-Voss A., Krueger G., Henighan T., Child R., Ramesh A., Ziegler D., Wu J., Winter C., Hesse C., Chen M., Sigler E., Litwin M., Gray S., Chess B., Clark J., Berner C., McCandlish S., Radford A., Sutskever I., Amodei D. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 1877–1901.
3. Wei J., Wang X., Schuurmans D., Maeda M., Edakunni J., Ku H., Le Q., Chi E. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 2022, vol. 35, pp. 24824–24837.
4. Kim J.-H., Nam J., Jo G. Computational Blueprints: Generating Isomorphic Mathematics Problems with Large Language Models. *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Suzhou, Association for Computational Linguistics, 2025, pp. 1389–1401.
5. Hong S., Zheng X., Chen J., Cheng Y., Zhang C., Wang Z., Yau S. K. S., Lin Z., Zhou L., Ran W., Xiao J., Wu M. MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework. *Proceedings of the International Conference on Learning Representations (ICLR)*. Vienna, OpenReview, 2024, pp. 1–26.
6. Zheng L., Chiang W.-L., Sheng Y., Zhuang S., Wu Z., Zhuang Y., Lin Z., Li Z., Li D., Xing E., Zhang H., Stoica I. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems*. Red Hook, Curran Associates, Inc., 2023, vol. 36, pp. 46561–46585.

7. Asenchik O. D., Jastrebov A. A. Classification of ways to vary laboratory tasks in educational disciplines of the information technology sphere. *Problemy sovremennogo obrazovaniya v tehničeskom vuze: materialy IX Mezhdunar. nauch.-metod. konf., Gomel', 23–24 okt. 2025 g.* [Problems of Modern Education in a Technical University: Materials of the IX International Scientific and Methodological Conference, Gomel', Oktober 23–24, 2025]. Gomel, 2025, pp. 62–64 (in Russian).
8. *LabGenerator*. *GitHub repository*. Available: <https://github.com/Olgasn/LabGenerator> (accessed 14 April 2026).

Поступила 17.04.2026 г.