

## СЕКЦИЯ II ТЕХНОЛОГИИ ОЦЕНКИ И КОНТРОЛЯ ЗНАНИЙ В ТЕХНИЧЕСКОМ ОБРАЗОВАНИИ

---

### КЛАССИФИКАЦИЯ СПОСОБОВ ВАРЬИРОВАНИЯ ЗАДАНИЙ ЛАБОРАТОРНЫХ РАБОТ ПО УЧЕБНЫМ ДИСЦИПЛИНАМ СФЕРЫ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

О. Д. Асенчик, А. А. Ястребов

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

В сфере IT-образования одной из ключевых задач является формирование у студентов навыков самостоятельного решения практических задач. Лабораторные работы занимают здесь особое место, так как позволяют не только закрепить теоретические знания, но и развить умение применения их на практике [1]. Однако их ценность может быть значительно снижена, если студенты выполняют задания несамостоятельно, зачастую прибегая к обмену готовыми решениями. Это снижает образовательный эффект и препятствует развитию профессиональной компетентности [2].

Одним из надежных способов решения этой проблемы является использование варьирования заданий лабораторных работ [3]. Варьирование позволяет создать широкий набор вариантов одного и того же задания или целого блока заданий, обеспечивая уникальность работы каждого студента и повышая вероятность ее самостоятельного выполнения. Это позволяет одновременно решать задачи борьбы с несамостоятельным выполнением работ, а также поддерживать высокий уровень вовлеченности студентов.

Однако создание цикла лабораторных работ по дисциплине с большим количеством вариантов является сложной и трудоемкой задачей. Появление и развитие больших языковых моделей (LLM) открывает новые возможности для автоматизации задач, связанных с генерацией текста, однако такие системы требуют задания содержательных вопросов, в частности осуществления способа варьирования вариантов.

Целью исследования является разработка классификации способов варьирования заданий лабораторных работ по учебным дисциплинам сферы информационных технологий. Такая классификация обеспечит преподавателей удобным инструментом при проектировании заданий на лабораторные и практические занятия и позволит формализовать процесс их генерации в автоматизированных образовательных системах.

В ходе предварительного анализа были выделены два базовых подхода к варьированию заданий в лабораторных работах. Первый способ заключается в том, что формулируется единое «мастер-задание», к которому подготавливается серия вариантов с варьированием предметной области или контекста, что позволяет проверять усвоение фундаментальных концепций в разных условиях. Второй предполагает отсутствие единого «мастер-задания» и достигает вариативности за счет изменения отдельных подзаданий, объединенных одной темой.

Эти подходы являются важной основой, однако они не отражают всего спектра возможных решений. Варьирование может затрагивать не только предметную область или формулировку подзаданий, но и входные данные, требования к представлению результатов, алгоритмы, используемые технологии и инструменты, а также критерии оценки. Для построения полноценной классификации необходимо учиты-

вать все перечисленные аспекты, что позволит создать более гибкую и практически применимую систему формирования заданий.

На основе анализа существующих подходов и изучения лабораторных работ по различным ИТ-дисциплинам выделены следующие способы варьирования заданий:

1. *Предметная область*: предполагает использование одной и той же структуры задания и набора подзаданий, но применение их к различным контекстам или сценариям. Например, задание на разработку алгоритма сортировки может быть дано в контексте различных задач (сортировка списка студентов, товаров, файлов).

2. *Наборы входных данных*: предоставление каждому студенту или группе студентов уникальных наборов входных данных для одной и той же задачи.

3. *Формат выходных данных (результатов)*: представление выходных данных или результатов в различных форматах. Например, для сохранения данных в файл – в форматах .txt, .csv или .bin, для визуализации полученных данных – презентация или отчет с различными типами графиков.

4. *Алгоритмические требования*: предложение реализовать одну и ту же функциональность, используя разные алгоритмические подходы или структуры данных. Например, при решении задачи поиска студентам может быть предложено использовать линейный поиск, бинарный поиск или хеширование.

5. *Ограничения на ресурсы и модули*: изменяются условия выполнения задания, такие как лимиты по времени, памяти, использование только определенных библиотек, требование реализации решения без использования встроенных коллекций языка программирования и др.

6. *Стек инструментов и технологий*: сохраняется общая задача, но меняется используемый стек технологий, т. е., например, языки программирования, среды разработки, фреймворки и др.

Перечисленные способы варьирования могут использоваться как по отдельности, так и в комбинации, что значительно расширяет число возможных уникальных вариантов лабораторных заданий. При выборе способов варьирования преподавателю необходимо определить цели и задачи лабораторной работы и убедиться, что выбранные способы варьирования действительно помогают их достичь.

Например, в дисциплине «Основы алгоритмизации и программирования» для лабораторной работы по изучению структур в языке программирования С можно использовать способы варьирования № 1, № 3, № 4 и № 5 – разные предметные области, алгоритмы сортировки данных, способы представления данных (в консоль в табличном виде или в файл формата .txt или .csv). В дисциплине «Базы данных» может быть применено варьирование предметной области и формата выходных данных.

Разработанная классификация способов варьирования заданий в лабораторных работах в сфере ИТ-образования позволяет повысить уникальность заданий и тем самым снизить риск несамостоятельного выполнения работ, стимулировать студентов к более глубокому пониманию материала, а также увеличить вовлеченность за счет разнообразия контекстов и технологий.

Кроме того, предложенная классификация может быть использована в автоматизированных системах генерации лабораторных заданий. В частности, она может лечь в основу механизма динамического составления промптов для больших языковых моделей, где параметры варьирования будут задавать конкретные условия задачи.

## Литература

1. Eckerdal, A. Learning programming practice and programming theory in the computer laboratory / A. Eckerdal, A. Berglund, M. Thune // European Journal of Engineering Education. – 2024. – № 49 (2). – P. 330–347.
2. Myronenko, S. Detecting Plagiarism in Student Assignments using Source Code Analysis // WSEAS Transactions on Computer Research. – 2024. – Vol. 12. – P. 367–376.
3. J. Vykořpal, P. Seda [et al.]. Preventing Cheating in Hands-on Lab Assignments. SIGCSE 2022: Proceedings of the 53rd ACM Technical Symposium on Computer Science Education, 2022. – P. 78–84.

**АВТОМАТИЗИРОВАННЫЙ КОНТРОЛЬ И ОЦЕНКА ЗНАНИЙ  
СТУДЕНТОВ ПО РЕЗУЛЬТАТАМ ОСВОЕНИЯ  
ПРОФЕССИОНАЛЬНЫХ ДИСЦИПЛИН СПЕЦИАЛЬНОСТИ  
«ЭЛЕКТРОЭНЕРГЕТИКА И ЭЛЕКТРОТЕХНИКА.  
ЭЛЕКТРОСНАБЖЕНИЕ» СРЕДСТВАМИ ЯЗЫКА  
ПРОГРАММИРОВАНИЯ PYTHON**

**М. С. Губатенко, Д. И. Зализный**

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

Современные средства компьютерной техники и программного обеспечения позволяют автоматизировать процесс проверки выполняемых студентами расчетов. Однако с целью выполнения такой проверки не обязательно обращаться к системам автоматизированного проектирования, использование которых требует платных лицензий. Для многих расчетов студенты используют табличный редактор, позволяющий в результате задания формул и взаимосвязей между ячейками выполнять многократно повторяющиеся расчеты с учетом изменений исходных данных или данных, получаемых в процессе расчета. Однако в связи с достаточным количеством студентов в группе, преподаватель при проверке результатов, полученных студентами, должен затратить достаточно большое количество времени. Например, проверка результатов расчета электрической нагрузки промышленного предприятия (цеха) может занимать до 10 минут, а при средней наполняемости группы, составляющей 20 человек, – более трех часов без перерыва. В связи с этим актуальным является вопрос разработки автоматизированной программы по проверке сходимости расчетов студентов с верными результатами.

Цель исследования – создание автоматизированной программы контроля расчетов студентов на основе доступных средств программирования.

Методы исследования – анализ доступных средств программного обеспечения для создания автоматизированной программы контроля расчетов студентов.

Исходя из проведенного анализа существующих средств программирования и возможной работы с файлами табличных редакторов, в качестве языка программирования выбран один из наиболее известных, современных и популярных языков программирования – Python [1].

Основные задачи, закладываемые в разрабатываемую программу:

1. Работа с фиксированной папкой, куда будут доставлены расчетные файлы студентов (например, D:\Результаты расчетов студентов).
2. Работа с фиксированной папкой, куда будет загружен файл преподавателя с номерами вариантов и верными результатами расчета (например, D:\Проверочный файл преподавателя\Результаты расчетов.xlsx).