

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра социальной информатики

ПРИМЕНЕНИЕ ПРОГРАММНЫХ АВТОГРЕЙДЕРОВ В ОБРАЗОВАНИИ

(автореферат бакалаврской работы)

студента 5 курса 531 группы
направления 09.03.03 - Прикладная информатика
профиль Прикладная информатика в социологии
Социологического факультета
Бисингалиева Рустама Иркеновича

Научный руководитель

профессор, доктор социологических наук, доцент _____ Н.И. Мельникова
подпись, дата

Зав. кафедрой

кандидат социологических наук, доцент _____ И.Г. Малинский
подпись, дата

Саратов 2025

ВВЕДЕНИЕ

Актуальность проблемы обусловлена стремительным развитием цифровых образовательных технологий и необходимостью повышения эффективности оценивания учебных достижений студентов. Особую значимость приобретает внедрение автоматизированных систем проверки заданий, таких как автогрейдеры, позволяющих оптимизировать процесс оценивания и обеспечить его объективность. В условиях растущих объемов проверяемых работ и увеличения числа обучающихся в высших учебных заведениях, автоматизация проверки становится важным инструментом поддержки образовательного процесса.

Современные системы автоматической проверки кода активно развиваются, предлагая новые возможности для повышения качества образования. Они способны не только проверять корректность выполнения заданий, но и предоставлять обратную связь, анализировать прогресс обучения и выявлять проблемные области в усвоении материала. При этом интеграция автогрейдеров с существующими образовательными платформами открывает широкие перспективы для создания гибридных моделей оценивания, сочетающих преимущества автоматизированных и традиционных методов проверки.

Однако внедрение систем автоматического оценивания сопряжено с рядом вызовов, включая обеспечение качества тестирования, адаптацию к различным дисциплинам и учет специфики образовательных программ. Необходимость решения этих задач требует проведения комплексных исследований по определению оптимальных сценариев применения автогрейдеров и разработки методических рекомендаций по их использованию в учебном процессе. В этой связи изучение возможностей и ограничений автоматизированных систем оценивания представляет собой актуальное направление исследований.

Степень разработанности проблемы автоматической проверки учебных заданий достаточно высока и охватывает несколько десятилетий исследований. Начиная с простейших пакетных систем проверки программ в 1960-х годах,

когда студенты загружали свои программы на центральный сервер для тестирования с predetermined входными данными, до современных сложных автогрейдеров, интегрированных в цифровую образовательную среду. Существенный вклад в развитие данной области внесли как зарубежные, так и отечественные исследователи.¹²³

Целью выпускной квалификационной работы является оценка границ применимости автогрейдеров при изучении программирования и анализе их эффективности в автоматической проверке и оценке качества учебных программных решений.

Для достижения поставленной цели необходимо решить следующие **задачи**:

1. Рассмотреть существующие подходы к автоматической оценке учащихся в образовании.

2. Проанализировать повестку дня в области будущего в автоматической оценке заданий.

3. Выполнить сравнительный анализ программных автогрейдеров и выбрать автогрейдер для работы.

4. Рассмотреть протоколы взаимодействия учебных платформ.

5. Осуществить интеграцию автогрейдера CodeGrade и LMS/LCMS Moodle.

6. Выполнить подготовку заданий по программированию на языке C для автогрейдера CodeGrade в Moodle

Объект данной работы — процесс автоматической оценки учебных заданий по программированию.

¹ Баранова Ю.С., Горшков Д.Н. Интеграция автономных систем оценивания с цифровыми образовательными ресурсами // Вестник компьютерных наук и информационных технологий. 2023. № 2. С. 71–82.

² Васильченко Л.И., Козлов Н.П. Повышение качества инженерного образования с использованием автогрейдеров // Инженерное образование XXI века. Материалы конференции. Москва, 2022. С. 211–219.

³ Галкин Г.М., Сергеев А.Б. Опыт внедрения автогрейдеров в университетском обучении информатике // Образование и наука. 2023. № 1. С. 93–101.

Предмет данной работы — программные средства автоматизированной оценки (автогрейдеры), их функциональные возможности, методы интеграции с системами управления обучением, а также особенности применения таких систем для проверки заданий по программированию на примере платформы CodeGrade и её взаимодействия с LMS Moodle.

Теоретическая выпускной квалификационной работы заключается в систематизации подходов к использованию автогрейдеров в образовательном процессе, уточнении их функциональных возможностей и выявлении ключевых факторов, влияющих на эффективность автоматической оценки учебных заданий по программированию.

Практическая значимость выпускной квалификационной работы заключается в возможности внедрения автогрейдера CodeGrade в образовательный процесс высших учебных заведений и онлайн-платформ для автоматизации проверки заданий по программированию, что позволяет сократить временные затраты преподавателей, повысить качество обратной связи и увеличить вовлеченность студентов в учебную деятельность.

Структура выпускной квалификационной работы представлена введением, тремя разделами, заключением, списком использованных источников и приложением.

В первом разделе «подходы к автоматической оценке учащихся в образовании» рассматриваются современные методы и средства автоматизированной проверки учебных заданий, анализируются тенденции их развития и выявляются основные перспективы и ограничения применения автогрейдеров в высшем образовании. Данный раздел посвящён обзору существующих технологий и инструментов, используемых в системах автоматического оценивания, а также анализу научных исследований, посвящённых данной проблеме.

Современное образование предлагает два основных направления проверки: традиционную ручную проверку и современные автоматизированные системы. В условиях цифровизации образовательного процесса всё большее

значение приобретают системы автогрейдинга, позволяющие сократить временные затраты на проверку заданий и повысить объективность оценки.

Автоматизированные методы делятся на несколько категорий, каждая из которых имеет свои особенности и области применения. Одним из ключевых подходов является статический анализ кода, который позволяет проверять программы без их выполнения. Например, при решении задачи преобразования строки в верхний регистр система может проверять наличие ключевого слова `upper()` и корректность форматирования кода. Такой подход широко используется для анализа стандартов кода с помощью инструментов Checkstyle⁴, ESLint⁵, Pylint⁶, которые оценивают наличие комментариев и общую структуру программы.

Следующим важным методом является тестирование – выполнение программ с различными входными данными. Иллюстрацией может служить проверка алгоритма сортировки массивов разного типа. Этот подход позволяет оценить функциональность программы в реальных условиях эксплуатации. Для тестов с множественным выбором или короткими ответами применяется метод сравнения с эталоном, обеспечивающий точное соответствие ответов заданным критериям.

Особое место занимает анализ текстов с помощью технологий обработки естественного языка (NLP). Этот метод позволяет автоматически проверять эссе и письменные работы на грамматику, ключевые слова и структуру. Например, при проверке эссе на тему «Социальные сети и их влияние на формирование общественного мнения», система может анализировать наличие определенных терминов (например, публичное мнение, социальный капитал, влияние медиа, дезинформация), оценивать логическую структуру текста (наличие введения, аргументированного основного текста, выводов), а также соответствие заданным объёмным требованиям.

⁴ <https://checkstyle.sourceforge.io/>

⁵ <https://eslint.org/>

⁶ <https://pylint.org/>

Так же используются комбинированные методы проверки, которые представляют собой наиболее эффективный подход, когда несколько методов используются параллельно. Например, программные задания могут проверяться сначала с помощью статистического анализа, затем тестированием, и завершаться визуальной проверкой преподавателем. Такой комплексный подход минимизирует вероятность упущений и позволяет учитывать сложные аспекты задач.

Однако важно отметить ограничения автоматизированных систем. Они испытывают трудности в понимании контекста, часто допускают ошибки при нестандартных решениях и сталкиваются с проблемами при оценке качества кода и творческих заданий. Поэтому подчеркивается необходимость сочетания автоматической проверки с ручной для достижения наилучших результатов.

Анализ повестки дня в области будущего автоматической оценки заданий показывает, что развитие систем автоматизированного оценивания сосредоточено вокруг трёх основных целей: повышения точности, обеспечения справедливости и усиления адаптивности оценочных механизмов. В условиях роста онлайн-обучения и цифровизации образования всё большее значение приобретает интеграция современных методов искусственного интеллекта — нейронных сетей, машинного обучения и технологий обработки естественного языка, — позволяющих глубже анализировать ответы учащихся и учитывать не только содержание, но и логику рассуждений.

Особое внимание в научном сообществе уделяется разработке механизмов персонализированной обратной связи, способных поддерживать индивидуальные образовательные траектории и стимулировать развитие критического мышления и самостоятельности у обучающихся. При этом остаются актуальными вопросы прозрачности алгоритмов, интерпретируемости результатов и предотвращения возможных дискриминационных эффектов, связанных с культурно-языковыми или социальными особенностями пользователей.

Важным аспектом является обратная связь от студентов. Системы могут стать еще более интерактивными, добавив возможность для студентов оставлять отзывы о предоставленных автогрейдером комментариях. Это помогло бы разработчикам улучшать алгоритмы и предоставлять более понятные и полезные результаты.

Анализ научных источников показывает, что тема автоматизации оценивания активно исследуется как в отечественной, так и в зарубежной практике и главной целью современных исследований становится создание интеллектуальных систем оценивания, сочетающих техническую эффективность с педагогической ценностью.

В втором разделе «Сравнительный анализ и выбор программного автогрейдера» рассматриваются различные аспекты выбора оптимальной системы автоматической проверки, что представляет собой комплексное исследование современных решений на рынке образовательных технологий.

В ходе этого исследования был проведен комплексный сравнительный анализ различных программных автогрейдеров, что позволило определить наиболее эффективные решения для автоматической проверки студенческих работ.

При выборе автогрейдера важно учитывать несколько ключевых факторов. Во-первых, функциональность системы - возможность поддержки различных типов задач, от программирования до анализа текстов и математических заданий. Во-вторых, производительность - система должна справляться с нагрузкой в условиях массовых онлайн-курсов (MOOCs). Например, при тестовой нагрузке в 1000 студентов время проверки одного задания должно составлять не более 5 секунд, а всех заданий - примерно 20 минут.

Особое внимание нужно уделить масштабируемости решений. Современные автогрейдеры, такие как CodeGrade, поддерживают параллельную обработку тысяч запросов через REST API, что особенно важно для крупных образовательных учреждений. При этом скорость проверки остается высокой даже при увеличении количества пользователей: для группы из 50 студентов

время проверки одного задания составляет менее 1 секунды, а всех заданий - около 30 секунд.

В результате проведенного анализа был выбран автогрейдер CodeGrade, который показал себя наиболее эффективным по нескольким параметрам. Эта система обеспечивает полную проверку кода, включая компиляцию (1-2 секунды), верификацию тестов (2-5 секунд) и формирование детального отчета об ошибках (1 секунда). Процесс занимает в среднем 8-10 секунд на одно задание.

CodeGrade предлагает богатые аналитические возможности для преподавателей, позволяя:

- ~ отслеживать динамику выполнения задач;
- ~ анализировать прогресс студентов в реальном времени;
- ~ получать подробную статистику по каждому учащемуся;
- ~ выявлять проблемные области обучения.

CodeGrade обладает рядом значительных преимуществ, которые делают его привлекательным выбором для образовательных учреждений. Система поддерживает различные языки программирования, включая Python, Java и C++, что обеспечивает гибкость при работе с разнообразными учебными курсами. Особенно ценной является возможность комбинировать автоматическую проверку с ручной, позволяя преподавателям дополнять машинную оценку собственными комментариями и замечаниями. Интерфейс платформы интуитивно понятен как для студентов, так и для преподавателей, что упрощает процесс адаптации к новому инструменту. Кроме того, разработчики CodeGrade предоставляют высокий уровень поддержки, помогая решать возникающие вопросы и технические проблемы.

Однако внедрение CodeGrade сопровождается определенными ограничениями. Одним из существенных недостатков является высокая стоимость лицензии, что может быть препятствием для некоторых учебных заведений. Система полностью зависит от интернет-подключения, что создает трудности для пользователей без стабильного доступа к сети. Перед началом

работы с каждым новым курсом требуется предварительная настройка, которая может быть достаточно сложной. Также стоит отметить необходимость наличия у преподавателей определенных навыков администрирования системы, что требует дополнительного времени и ресурсов на обучение педагогического состава.

Интеграция CodeGrade с Moodle через LTI и REST API предоставляет значительные преимущества. Стандартизированные протоколы обеспечивают эффективный обмен данными между системами, автоматическое обновление журнала оценок и повышают производительность учебного процесса.

Важно отметить, что выбор автогрейдера должен основываться на конкретных потребностях учебного заведения. Для небольших групп учащихся могут подойти более простые сервисы, такие как TestDriven.io, тогда как масштабные курсы требуют более сложных решений.

Автоматизированные системы грейдинга имеют потенциал не только для проверки знаний, но и для развития критического мышления учащихся. Они предоставляют подробную обратную связь, показывая, где именно студент совершил ошибки, и предлагая ресурсы для их исправления. В социальном и образовательном аспектах такие системы способствуют снижению нагрузки на преподавателей, что позволяет им уделять больше времени творческим задачам и индивидуальной поддержке учеников, а автоматизация второстепенных элементов преподавания открывает дорогу к более интерактивным и вдохновляющим урокам.

В третьей главе выпускной квалифицированной работы рассматриваются протоколы взаимодействия учебных платформ, интеграции автогрейдера CodeGrade и LMS/LCMS Moodle, а также представлены задания по программированию на языке C для автогрейдера CodeGrade в Moodle.

Взаимодействие между системой управления обучением Moodle и внешними приложениями, такими как автогрейдер CodeGrade, реализуется через стандартизированные протоколы обмена данными таких как LTI и REST API. Эти протоколы обеспечивают корректность, безопасность и эффективность

передачи информации между системами, а также позволяют интегрировать современные образовательные технологии в привычную среду обучения.

Одним из ключевых стандартов, используемых для интеграции, является LTI (Learning Tools Interoperability) — протокол, разработанный консорциумом IMS Global Learning. LTI позволяет легко подключать сторонние обучающие инструменты к платформе Moodle, обеспечивая единую точку входа, защиту данных и передачу контекстуальной информации о студентах и заданиях. Например, когда студент выполняет задание в Moodle, система может перенаправить его на интерфейс CodeGrade, сохранив при этом авторизационные данные и контекст задания. Это делает процесс использования автогрейдера максимально прозрачным и удобным как для преподавателей, так и для студентов.

Для взаимодействия с сервером Moodle через HTTP-запросы используется программный интерфейс REST API. REST API предоставляет гибкий способ обмена информацией: отправка заданий на проверку, получение результатов, обновление журналов успеваемости и другие операции. Благодаря использованию форматов JSON или XML для передачи данных, REST API обеспечивает высокую степень масштабируемости, что особенно важно для крупных университетов, где одновременно могут обрабатываться тысячи запросов.

Использование этих протоколов позволяет не только передавать информацию, но и организовывать обратную связь в режиме реального времени. Например, после автоматической проверки задания в CodeGrade, результаты могут быть мгновенно отправлены обратно в Moodle, где они отобразятся в журнале успеваемости, а студент получит комментарии и рекомендации по улучшению решения.

Архитектура интеграции систем включает два основных компонента: сервер Moodle и сервер CodeGrade, которые связаны между собой через коммуникационные каналы, такие как REST API и LTI. Каждый из этих компонентов выполняет свою функцию:

Сервер Moodle выступает центральной образовательной платформой, где размещаются курсы, учебные материалы, задания и журналы успеваемости. Он обеспечивает хранение данных о студентах, преподавателях, курсах и результатах обучения. Также он предоставляет интерфейсы для подключения внешних приложений, в том числе через REST API и LTI.

Сервер CodeGrade представляет собой ядро автогрейдинга, которое отвечает за прием кода, его выполнение, запуск тестов и формирование результатов. Он поддерживает несколько языков программирования, таких как Python, Java, C++ и другие, и предоставляет аналитические инструменты для оценки качества кода, выявления ошибок и выдачи рекомендаций.

Взаимодействие между Moodle и CodeGrade организовано следующим образом:

1. Запуск задания: студент загружает решение задачи через интерфейс Moodle. Задание передается в CodeGrade через LTI или API.

2. Автоматическая проверка: CodeGrade компилирует полученный код, запускает предопределенные тесты, анализирует результаты и формирует оценку с комментариями.

3. Возврат результата: После завершения проверки результаты отправляются обратно в Moodle через Webhook или API, где они отображаются в журнале успеваемости, а студент получает подробную обратную связь.

Для обеспечения безопасности и корректности обмена данными используется протокол OAuth 2.0, который управляет авторизацией и доступом к ресурсам. Общая схема архитектуры выглядит следующим образом: Moodle ↔ API Gateway ↔ CodeGrade

Такая структура обеспечивает надежное и масштабируемое взаимодействие, что особенно важно при работе с большими группами студентов. Кроме того, система предусматривает синхронизацию только необходимых данных между Moodle и CodeGrade, таких как информация об

оценках, статусе выполнения заданий и комментариях, что снижает нагрузку на серверы и повышает производительность.

В рамках практической части были разработаны и внедрены шесть учебных заданий, охватывающих ключевые темы курса «Управляющие структуры», такие как работа с символами, вычисление функции с условиями, мультиветвление и использование циклов. Каждое задание было тщательно спроектировано с учётом методических целей преподавания программирования и требований к автоматизированной проверке. В процессе подготовки заданий предусматривались следующие компоненты: шаблон кода для студентов, эталонное решение, набор тестовых кейсов, охватывающих типовые и граничные случаи, а также система оценивания, основанная на прохождении тестов и анализе качества кода.

Для обеспечения корректной работы автогрейдера в среде CodeGrade была проведена предварительная настройка заданий, включая определение параметров компиляции, выбор используемых компиляторов (например, GCC), формирование входных и выходных данных для тестирования, а также настройку метрик оценки – таких как количество пройденных тестов, соблюдение стандартов кодирования, использование управляющих конструкций и эффективность решения. Для каждого задания были подготовлены не только положительные сценарии проверки, но и тесты, моделирующие ошибочные или крайние случаи, чтобы гарантировать надёжность и полноту оценки.

Интеграция системы CodeGrade с образовательной платформой Moodle проводилась через протокол LTI. Это позволило организовать единое информационное пространство, в котором обучающиеся могут получать задания, отправлять свои решения и видеть результаты проверки непосредственно в интерфейсе Moodle. Преподаватели, в свою очередь, имеют возможность управлять заданиями, просматривать аналитику успеваемости студентов и при необходимости вносить коррективы в автоматически сгенерированные оценки.

Процесс проверки заданий был полностью настроен и автоматизирован: после загрузки решения студентом, система CodeGrade запускала его компиляцию, выполняла тестирование на основе заранее заданных кейсов, формировалась отчетность с указанием успешности выполнения каждого теста, а также предоставлялась обратная связь по найденным ошибкам. Полученные данные о результатах проверки синхронизировались с Moodle, где они становились доступны как обучающимся, так и преподавателям.

Было отмечено удобство аналитики, которую предоставляла система: можно было отслеживать динамику успеваемости каждого студента, выявлять темы, вызывающие наибольшие трудности, и прогнозировать вероятность успешной сдачи курса. Интеграция CodeGrade с Moodle заняла менее часа, при этом большинство задач, таких как синхронизация списка студентов и заданий, были автоматизированы, что существенно снизило нагрузку на ИТ-персонал.

Однако автогрейдеров есть ряд ограничений использования. Одним из них является необходимость предварительной настройки тестов и критериев оценки, что требует времени и внимания со стороны преподавателя. Также система имеет ограниченные возможности персонализированной обратной связи, особенно при нетиповых или творческих решениях заданий. В некоторых случаях автогрейдер неправильно может неправильно интерпретировать формат вывода в нестандартных практиках, что потребует донастройки «проверочных тестов».

Интеграция автогрейдера CodeGrade с системой Moodle показала высокую эффективность в условиях образовательной среды. Система позволяет значительно ускорить процесс проверки заданий, повысить качество обучения за счет детализированной и моментальной обратной связи и увеличить вовлеченность студентов. Однако успешное внедрение системы требует тщательной подготовки курсов, настройки под конкретные учебные программы и дополнительных финансовых и временных ресурсов на начальном этапе. В конечном итоге, автоматизация оценивания должна рассматриваться не как замена преподавателя, а как мощный инструмент, поддерживающий образовательный процесс и позволяющий сделать его более современным,

доступным и ориентированным на развитие профессиональных компетенций студентов.

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе показана актуальность и перспективность использования автоматизированных систем оценки – автогрейдеров – в образовательной практике. Автогрейдеры позволяют значительно сократить время, затрачиваемое преподавателями на рутинную проверку заданий, а также делает оценивание более объективным и обоснованным. Это особенно важно при массовом обучении, дистанционных курсах и цифровой трансформации образования.

В ходе выпускной квалификационной работы были проанализированы различные подходы к автоматической оценке: статический анализ кода, тестирование с входными данными, сравнение с эталонами и анализ текста с помощью технологий NLP. Выявлено, что у каждой методики есть свои сильные стороны и ограничения. Их комбинированное применение повышает общее качество оценивания.

Анализ повестки дня в области будущего автоматической оценки заданий показывает, что ключевые исследования и разработки сосредоточены на повышении точности, справедливости и адаптивности систем оценивания. Особое внимание уделяется интеграции современных методов искусственного интеллекта, таких как нейронные сети и машинное обучение, для более глубокого анализа ответов учащихся и предоставления персонализированной обратной связи. Не менее важными направлениями остаются обеспечение прозрачности алгоритмов и предотвращение дискриминационных эффектов. Основная цель — создание систем, которые не только эффективно и объективно оценивают знания, но и способствуют развитию критического мышления и самостоятельности обучающихся.

В ходе сравнительного анализа программных решений для автоматической проверки заданий автогрейдер CodeGrade был выбран в качестве наиболее

подходящего инструмента благодаря гибким настройкам, интуитивно понятному интерфейсу и возможности глубокой интеграции с образовательной платформой Moodle. Было изучено взаимодействие системы с LMS, успешно реализована интеграция через протокол LTI 1.3, а также учтены практические аспекты развертывания: настройка API, подготовка учебного контента и обеспечение корректной передачи результатов оценивания.

Практическая реализация показала, что внедрение CodeGrade в образовательную среду Moodle позволяет эффективно организовать процесс автоматической проверки практических заданий по программированию на языке C. Разработанные задания охватывали ключевые темы курса «Управляющие структуры», включая работу с символами, вычисление функций с условиями, мультиветвление и использование циклов. Для каждого задания были созданы эталонные решения, шаблоны кода для студентов, а также набор тестовых кейсов, охватывающих типовые, граничные и ошибочные ситуации. Тестирование было реализовано с использованием регулярных выражений и дифференцированной системы баллов, что повышает объективность оценки и позволяет предоставлять обучающимся содержательную обратную связь, направленную на улучшение качества программного кода.

Интеграция автогрейдера с Moodle обеспечила целостность образовательного процесса: обучающие выполняют задания в едином интерфейсе, а результаты автоматически передаются в журнал успеваемости, что упрощает отслеживание прогресса и контроль со стороны преподавателя.

Однако, несмотря на очевидные преимущества, автоматизированные системы оценки имеют ряд ограничений. К ним относится сложность анализа креативных и нетипичных решений, значительные временные затраты на начальном этапе настройки системы, а также вопросы этики и прозрачности используемых алгоритмов. Эти проблемы могут быть смягчены за счёт применения гибридного подхода, сочетающего автоматическую оценку с элементами ручной проверки.

Разработанные задания и пример интеграции могут быть внедрены в преподавание программирования, а также адаптированы под другие

образовательные программы. Внедрение автогрейдеров, таких как CodeGrade, способствует повышению мотивации обучающихся за счёт предоставления мгновенной и содержательной обратной связи. В свою очередь, автоматизация проверки снижает нагрузку на преподавателей и делает обучение более доступным в глобальном масштабе, особенно в условиях онлайн- и смешанного обучения, где важна эффективность и масштабируемость образовательных процессов.

Перспективы дальнейших исследований в области автоматической оценки учебных заданий связаны с несколькими ключевыми направлениями. Одним из них является разработка адаптивных систем оценивания на основе машинного обучения, способных учитывать индивидуальные особенности обучающихся и динамически подстраивать задания под уровень подготовки студента. Также актуальным остаётся расширение возможностей автогрейдеров за счёт интеграции с различными образовательными платформами, такими как Canvas, Blackboard, EdX, Coursera и другими, что повысит их доступность и применимость в разных образовательных экосистемах.

Ещё одной областью развития являются интерактивные учебные материалы, которые могут помочь студентам не просто получать результат проверки, но и глубже понимать свои ошибки. К таким материалам относятся: пошаговые объяснения выполнения кода, интерактивные трассировщики программ, визуализация тестовых сценариев, рекомендации по исправлению ошибок, а также мини-обучающие модули, активируемые при выявлении типичных затруднений. Эти элементы позволят превратить процесс получения обратной связи в полноценную обучающую активность. Таким образом, автоматизация оценивания становится важным инструментом трансформации образования, способствуя его адаптации к современным вызовам.