

в работе учителей физики как методическое пособие; при разработке стандартов интеграции ИИ в образовательный процесс. ИИ не заменяет преподавателя, а является инструментом для усиления, расширения возможностей человеческого интеллекта. Достигнута возможность использования ИИ как инструмента формирования физического мышления за счет создания методически строгой системы промпта.

Литература

1. Руководство по использованию генеративного искусственного интеллекта в образовании и научных исследованиях: издание ЮНЕСКО. – URL: <https://aspnet-unesco.ru/-prod/files/doc/event/c4d192913ca89aa65b274ac99c2b4214.pdf> (дата обращения: 15.04.2025).
2. Перечень сервисов искусственного интеллекта / Глав. информ.-аналит. центр М-ва образования Респ. Беларусь (ГИАЦ). – URL: https://akademy.by/images/Obuchenie/ИИ/Сервисы_искусственного_интеллекта_перечень.pdf (дата обращения: 15.04.2025).

АНАЛИЗ ПОТЕРЬ ДАННЫХ В ЦЕПОЧКЕ МЕЖСИСТЕМНОГО ОБМЕНА ФОРМАТАМИ

В. В. Лукашевич, А. А. Скуковская

*Полоцкий государственный университет имени Евфросинии Полоцкой,
г. Новополоцк, Республика Беларусь*

Рассмотрена проблема потери и искажения данных при передаче информации между различными программными системами. Анализируется типичная цепочка обмена: Excel → CSV → Python → JSON → веб-форма → база данных. Оценены этапы, на которых наиболее вероятны ошибки, и предложены рекомендации по их минимизации.

Ключевые слова: обмен данными, совместимость форматов, CSV, JSON, Excel, Python, база данных.

ANALYSIS OF DATA LOSS IN INTERSYSTEM FORMAT EXCHANGE CHAIN

V. V. Lukashevich, A. A. Skukovskaya

*Euphrosyne Polotskaya State University of Polotsk, Novopolotsk,
Republic of Belarus*

This article examines the issue of data loss and distortion during information exchange between various software systems. A typical data exchange chain is analyzed: Excel → CSV → Python → JSON → web form → database. The stages most prone to errors are evaluated, and recommendations for minimizing such risks are proposed.

Keywords: data exchange, format compatibility, CSV, JSON, Excel, Python, database.

В условиях цифровой трансформации все чаще возникает необходимость обмена данными между разнородными системами – от офисных приложений до веб-сервисов и баз данных. Несмотря на наличие стандартов, на практике при конвертации данных часто происходят потери или искажения. Особенно уязвимыми оказываются текстовые поля, даты, числа с плавающей точкой и специальные символы. Цель работы – выявить критические точки в цепочке обмена и проанализировать причины возникающих ошибок.

Типичная цепочка обмена начинается с формирования исходных данных в Microsoft Excel. Затем они экспортируются в CSV, считываются скриптом на Python, преобразуются в формат JSON и передаются через веб-форму, например, с исполь-

зованием REST API. Финальным этапом является сохранение информации в реляционной базе данных, такой как PostgreSQL. На каждом шаге возможны потери или искажения, что требует внимательного анализа. Для проверки использованы тестовые данные, представленные на рис. 1.

	1	2	3	4
1	ФИО	ДАТА РОЖДЕНИЯ	СРЕДНИЙ БАЛЛ	КОММЕНТАРИЙ
2	Иванов И.И.	01.03.2004	8,75	Студент, "отличник", №123

Рис. 1. Входные данные

Экспорт из Excel в CSV – первый и часто самый критичный этап. Excel сохраняет не только значения, но и метаданные: формулы, форматирование, типы данных, локальные настройки. При экспорте в CSV все это теряется – остаются только строки. Русскоязычные версии Excel используют кодировку Windows-1251 и точку с запятой как разделитель, что противоречит стандарту RFC 4180 (UTF-8 и запятая). Это вызывает проблемы при открытии файла в Linux-системах или Python-скриптах: кириллица превращается в иероглифы, структура таблицы нарушается. Даже корректное открытие в другом Excel создает иллюзию совместимости, которая рушится при передаче данных.

Чтение CSV в Python (обычно через pandas) сопровождается рисками: если не указать явно encoding и sep, библиотека делает догадки, часто ошибочные. Кодировка cp1251, прочитанная как UTF-8, приводит к некорректному отображению символов, а неверный разделитель превращает строку в одно поле. Особенно уязвимы текстовые поля с запятыми, кавычками, переводами строк. Excel не всегда строго следует стандарту CSV, что приводит к смещению колонок. Проблемы возникают и с датами: Excel хранит их как число дней с 01.01.1900 (с ошибкой 29 февраля 1900). Если дата не отформатирована как текст, Python прочитает ее как число, теряя временную информацию.

Преобразование в JSON добавляет сложности: формат поддерживает ограниченный набор типов. Все остальное (datetime, Decimal, множества) требует явной сериализации. Даже при корректной сериализации возможны потери: числа с плавающей точкой могут измениться из-за IEEE 754. JSON требует экранирования спецсимволов – без этого документ становится невалидным, и парсер его отклоняет.

Передача через веб-интерфейс сопровождается валидацией. Фреймворки (Django, Flask, Express.js) используют схемы (Pydantic, JSON Schema), задающие требования к типам, форматам, длине. Например, дата в формате DD.MM.YYYY может быть отклонена, если ожидается ISO 8601. Текстовые поля проходят санитизацию: HTML-теги и опасные символы экранируются. Фраза «Студент "отличник" № 123» может превратиться в «Студент отличник №123».

Сохранение в БД также несет риски. Вставка строки "8,75" в NUMERIC может вызвать ошибку из-за запятой. Строка длиной 60 символов в поле VARCHAR(50) будет усечена. Даты без временной зоны могут интерпретироваться по локали СУБД, вызывая расхождения. Некоторые СУБД (например, MySQL) могут «угадывать» дату, подставляя 0000-00-00 вместо невалидной.

Таким образом, каждый этап обмена данными – от Excel до БД – несет риск потерь и искажений. Особенно уязвимы экспорт и веб-передача, где отсутствие единых стандартов и валидации приводит к систематическим ошибкам. Только соблюдение открытых стандартов (UTF-8, ISO 8601, RFC 4180), контроль типов и кодировок, а

также сквозная валидация и логирование обеспечивают надежную передачу данных между системами. Данные, дошедшие до БД, показаны на рис. 2.

FULL_NAME	BIRTH_DATE	AVG_SCORE	COMMENT
Иванов И.И.	NULL	NULL	Студент, отличник, №123

Рис. 2. Конечный результат

Цепочка обмена данными между системами – это последовательность трансформаций, каждая из которых несет риск потери или искажения информации. Наиболее уязвимыми этапами являются экспорт из Excel в CSV и передача через веб-интерфейс. Для минимизации ошибок рекомендуется:

- использовать UTF-8 с BOM при экспорте CSV;
- явно указывать разделители и кодировки при чтении;
- применять стандартизированные форматы (например, ISO 8601 для дат);
- валидировать данные на каждом этапе передачи.

Только системный подход к совместимости форматов позволяет обеспечить целостность и достоверность данных в межсистемном взаимодействии.

Литература

1. McKinney, W. Python for Data Analysis. – O'Reilly Media, 2022.
2. Bray, T. (Ed.). The JavaScript Object Notation (JSON) Data Interchange Format. – RFC 8259, 2017.
3. Microsoft. Excel Specifications and Limits. – Документация Microsoft, 2023.

ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ К ПРОГРАММЕ ДЛЯ УЧЕТА КУРАТОРСКОЙ ГРУППЫ

М. А. Посох, К. Ю. Булохов, А. А. Скуковская

*Полоцкий государственный университет имени Евфросинии Полоцкой,
г. Новополоцк, Республика Беларусь*

Рассмотрены функциональные требования к программному обеспечению для учета кураторской группы. Анализируются ключевые компоненты системы, включая личные кабинеты студента и куратора, механизмы регистрации, редактирования профиля, ведения посещаемости, подачи справок и получения уведомлений. Особое внимание уделено возможностям куратора по учету воспитательных мероприятий, научной активности студентов и интеграции с медиаканалами.

Ключевые слова: учет кураторской группы, личный кабинет куратора, учет посещаемости, воспитательная деятельность, научная активность студентов, цифровая отчетность.

FUNCTIONAL REQUIREMENTS FOR SOFTWARE TO MANAGE A CURATORIAL GROUP

M. A. Posakh, K. Y. Bulokhov, A. A. Skukovskaya

*Euphrosyne Polotskaya State University of Polotsk, Novopolotsk,
Republic of Belarus*

This article examines the functional requirements for software designed to manage a curatorial group. Key system components are analyzed, including personal accounts for students and curators, mechanisms for registration, profile editing, attendance tracking, submission of documents, and notifications. Special attention is given to the curator's capabilities in tracking