

Вайб-кодинг в образовании: генерация тренажера слепой печати средствами искусственного интеллекта

Пайдершев Алексей Константинович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье рассматриваются перспективы применения генеративных нейросетей для конструирования образовательных веб-интерфейсов на примере разработки интерактивного тренажера слепой печати в среде Qwen Studio. Опора на базовые инструменты браузера по перехвату событий клавиатуры обеспечивает создание независимого, легковесного приложения без необходимости внедрения сторонних фреймворков и тяжелых зависимостей. Представленный метод автоматизированной кодогенерации посредством точного промпт-инжиниринга кардинально оптимизирует производственный цикл, передавая рутинные задачи алгоритмического проектирования искусственному интеллекту. Внедрение строгих математических метрик для вычисления скорости и точности набора текста служит надежным критерием оценки того, насколько эффективно языковая модель справляется с синхронной обработкой пользовательских действий, сложным динамическим обновлением объектной модели документа и поддержанием моментального визуального отклика интерфейса.

Ключевые слова: Qwen Studio, нейросети, промпт-инжиниринг, слепая печать, вайб-кодинг.

Vibe-coding in education: generating a touch typing simulator using artificial intelligence

Paydershev Alexey Konstantinovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article discusses the prospects of using generative neural networks to construct educational web interfaces, using the example of developing an interactive blind typing simulator in the Qwen Studio environment. By relying on the browser's basic tools for capturing keyboard events, it is possible to create an independent, lightweight application without the need for third-party frameworks or heavy dependencies. The presented method of automated code generation through precise reverse engineering significantly optimizes the development process by delegating routine algorithmic design tasks to artificial intelligence. The implementation of strict mathematical metrics for calculating the speed and accuracy of typing serves

as a reliable criterion for evaluating how effectively a language model handles synchronous processing of user actions, complex dynamic updates to the document object model, and maintaining instant visual response to the interface.

Keywords: Qwen Studio, neural networks, prompt-engineering, blind typing, viber coding.

1 Введение

1.1 Актуальность

Актуальность исследования продиктована растущей потребностью в быстром создании специализированных образовательных инструментов и интерактивных веб-тренажеров. Традиционная разработка подобных приложений требует от специалистов глубокого понимания принципов управления интерфейсными состояниями, синхронной обработки пользовательских событий и сложного взаимодействия с объектной моделью документа. Применение стандартных современных фреймворков для решения таких локальных задач часто оказывается технически избыточным, что замедляет реализацию проектов и ограничивает возможности авторов образовательного контента в оперативной подготовке персонализированных цифровых платформ.

Внедрение больших языковых моделей в процесс создания программного обеспечения предлагает принципиально иной подход к решению обозначенной проблемы. Делегирование написания программного кода искусственному интеллекту позволяет перенести фокус внимания исключительно на проектирование логики обучения и пользовательского опыта. Изучение того, насколько эффективно детализированный промпт-инжиниринг справляется с реализацией математических вычислений и обеспечением моментального визуального отклика интерфейса в реальном времени, имеет высокую практическую значимость. Описанный метод доказывает возможность мгновенного развертывания сложных интерактивных систем, минуя ресурсоемкий этап классического программирования и делая веб-разработку доступной для более широкого круга специалистов.

1.2 Обзор исследования

Фундаментальные принципы конструирования интерактивных браузерных продуктов на базе HTML5 подробно раскрыты в исследованиях С.В. Рогатина и В. А. Антошкина [1]. Вопросы интеграции искусственного интеллекта в цифровую педагогику поднимает А. Б. Рахмонов, описывая практическую пользу современных форматов машинного взаимодействия для образовательного процесса [2]. Техническую сторону алгоритмической генерации развивает И. В. Елманов, который настаивает на строгом инженерном контроле и предлагает использовать концепцию Spec Driven Development для проверки надежности сгенерированного кода [3]. Применимость генеративных моделей за пределами программирования подтверждают работы Д. Р. Мухаметова, изучающего влияние ИИ на

формирование политического образа будущего посредством маркетинговых инструментов [4], а также труд А. Д. Степанова, Л. А. Богданова, Д. К. Шишковой и А. Г. Кутихина, где доказана результативность нейросетевых алгоритмов при решении узкоспециализированных медицинских задач в кардиологии [5].

1.3 Цель исследования

Цель исследования – оценка эффективности применения структурированного промпт-инжиниринга и вычислительных возможностей больших языковых моделей (на примере Qwen Studio) для автоматизированного проектирования автономных веб-приложений.

2 Материалы и методы

В основе исследования лежит методология, где классическое программирование заменено автоматической генерацией кода с помощью точного промпт-инжиниринга. Процесс начинается с утверждения строгих архитектурных ограничений: веб-приложение конструируется в виде единственного файла без подключения внешних зависимостей, используя исключительно базовые функции браузера для отслеживания пользовательского ввода.

При составлении запроса к искусственному интеллекту главный акцент делается на синхронную обработку данных. В текстовой инструкции детально прописываются алгоритмы перехвата нажатий клавиш, формулы для вычисления скорости и точности набора, а также правила динамического изменения цвета текста. Это гарантирует мгновенную визуальную реакцию интерфейса на каждую ошибку или правильное действие. Итоговая сборка вычислительной логики, визуального оформления и навигации в готовый автономный документ выполняется мощностями платформы Qwen Studio. Подобный метод позволяет разрабатывать полноценные интерактивные образовательные инструменты с минимальными затратами ручного труда.

3 Результаты

Первый шаг исследования – написание промта для генерации кода.

Промт: Act as a Senior Frontend developer. Your task is to write a complete, ready-to-run code for the «Тренажер слепой печати» web application. The application must be implemented in a single file and work locally in the browser. No third-party libraries are allowed.

Architectural requirements:

1. Implement a tab system with two tabs: «Тренажер» (active by default) and «Обучение.» Use pure JS for switching between tabs.

2. The «Обучение» tab: a neatly formatted text with the rules of blind typing and a `<img>` tag with a link to 'https://placeholder.co/800x400/2b2b36/4caf50?text=Keyboard+layout'.

Requirements for the «Тренажер» tab:

- Add the «Свой текст» functionality: place a compact block with a text field (placeholder: «Вставьте свой текст сюда...») and a «Загрузить текст» button above the main game field. When you click the button, the text in the field should become the current task, and the statistics and timer should be reset.

- Typing logic: use «keydown». The timer starts when you press the key for the first time.

- By default (if your text is not loaded), use the built-in array of 3 random sentences in Russian.

- Real-time statistics: WPM (formula: (number of correct characters / 5) / time in minutes) and Accuracy (%).

- Color indication: correct character is green, error is red, and the current character (waiting for input) is highlighted with an underline or a contrasting background.

- Leave the «Начать заново» button below the text.

- Make sure to block the standard page scrolling when you press the space bar during printing.

Style requirements:

- Modern, minimalist design. Dark theme.

- The block for loading your text should not take up too much space, and it should be aesthetically pleasing and visually separated from the printing area.

- The code should be clean and commented in Russian.

Display the full code in a single block. The application's language is Russian.

(Действуй как Senior Frontend разработчик. Твоя задача – написать полный, готовый к запуску код веб-приложения «Тренажер слепой печати». Приложение должно быть реализовано в одном файле и работать локально в браузере. Никаких сторонних библиотек.

Архитектурные требования:

1. Реализуй систему вкладок: «Тренажер» (активна по умолчанию) и «Обучение». Переключение на чистом JS.

2. Вкладка «Обучение»: аккуратно сверстаный текст с правилами слепой печати и тег `<img>` со ссылкой 'https://placeholder.co/800x400/2b2b36/4caf50?text=Схема+клавиатуры'.

Требования к вкладке «Тренажер»:

- Добавь функционал «Свой текст»: над основным игровым полем размести компактный блок с текстовым полем (плейсхолдер: «Вставьте свой текст сюда...») и кнопкой «Загрузить текст». При нажатии на кнопку, текст из поля должен становиться текущим заданием, а статистика и таймер – сбрасываться.

- Логика набора: используй «keydown». Таймер запускается при первом нажатии.

- По умолчанию (если свой текст не загружен) используй встроенный массив из 3 случайных предложений на русском языке.

- Статистика в реальном времени: WPM (формула: (кол-во правильных символов / 5) / время в минутах) и Точность (%).

- Цветовая индикация: правильный символ – зеленый, ошибка – красный, текущий символ (ожидающий ввода) – выделен подчеркиванием или контрастным фоном.

- Под текстом оставь кнопку «Начать заново».

- Обязательно блокируй стандартную прокрутку страницы при нажатии пробела во время печати.

Требования к стилю:

- Современный, минималистичный дизайн. Темная тема.

- Блок загрузки своего текста не должен занимать слишком много места, сделай его эстетичным и визуально отделенным от зоны печати.

- Код должен быть чистым и прокомментированным на русском языке.

Выведи полный код в одном блоке. Язык приложения русский.)

Результат

генерации:

<https://drive.google.com/file/d/1lBzrrl2rtQ3EEPwYV-9qQPvs79PKjhzT/view?usp=sharing>

В результате применения описанной методологии был успешно сгенерирован полностью функциональный прототип интерактивного тренажера слепой печати (рис. 1). Искусственный интеллект сформировал готовый к запуску документ, не требующий предварительной ручной отладки. Модель строго соблюла заданные архитектурные ограничения, реализовав интерфейс, систему навигационных вкладок и вычислительную логику исключительно средствами нативного JavaScript, без привлечения сторонних библиотек.

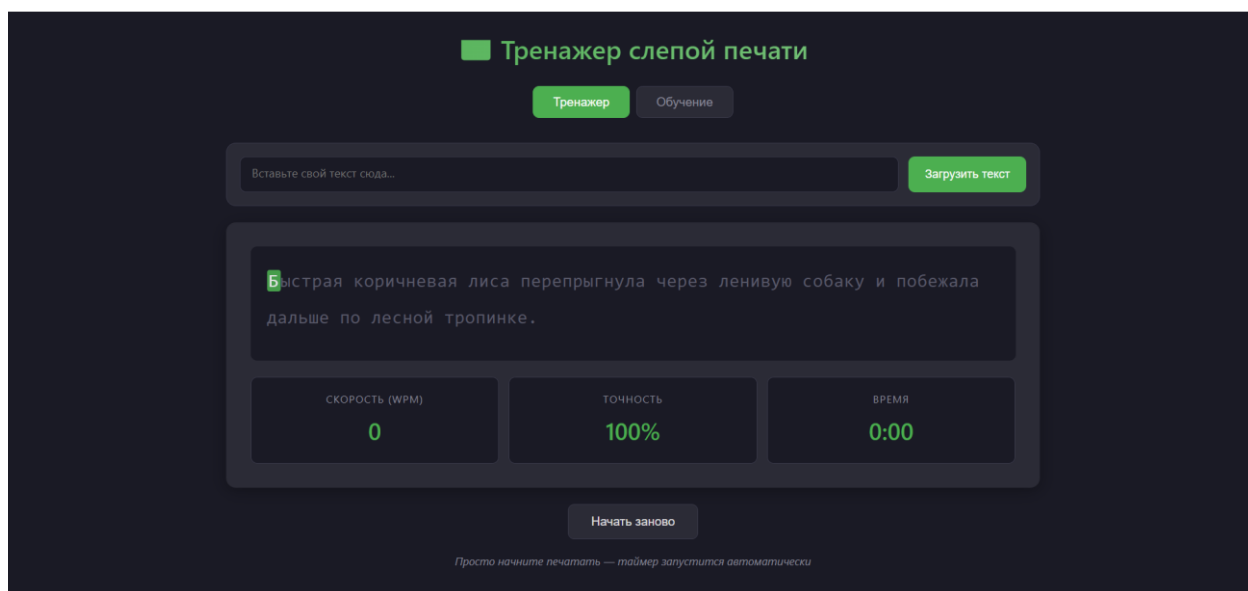


Рисунок 1. Интерфейс приложения

По умолчанию в программу защиты три тестовых предложения на русском языке для тренировки. Однако в приложении существует функция добавления своего текста, при использовании которой, тестовые предложения заменяются на текст, добавленный пользователем (рис. 2).

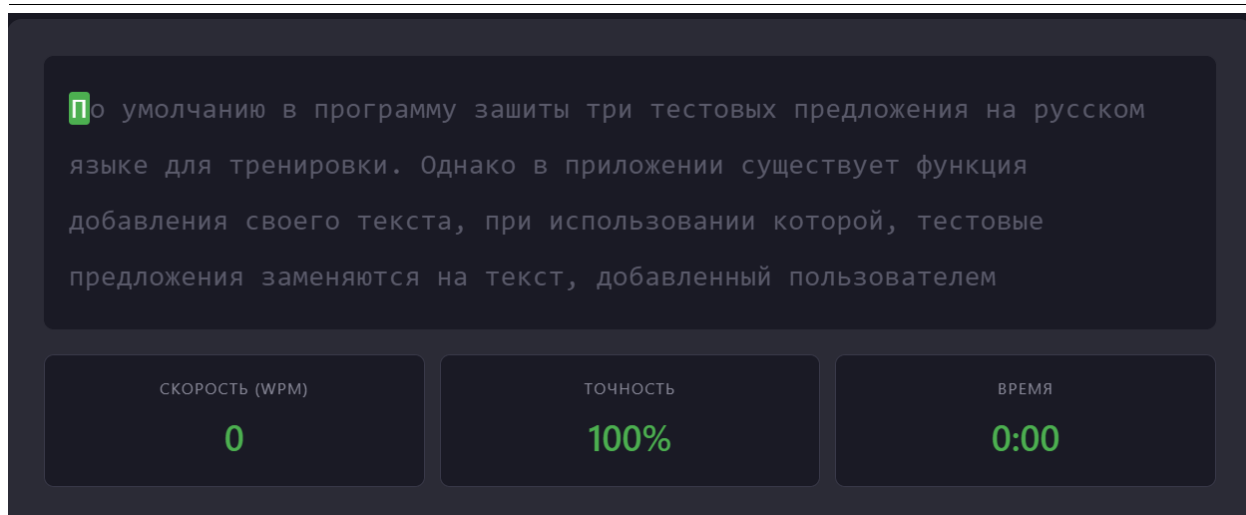


Рисунок 2. Добавление текста

Также в приложение встроена индикация набранных символов, где (рис. 3):

- Серый – ожидающие символы
- Зелёный с пульсацией – текущий символ
- Зелёный – правильно набранные
- Красный с подсветкой фона – ошибки

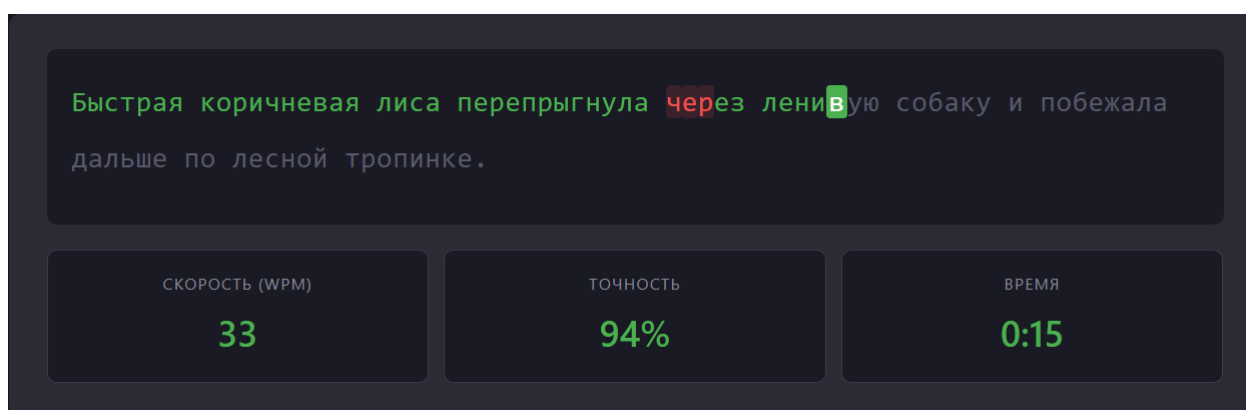


Рисунок 3. Индикация набранных символов

Таймер запускается при наборе первого символа на клавиатуре. Точность (F) определяется по формуле (x – кол-во правильно набранных символов, y – кол-во набранных символов):

$$F = \frac{x}{y} \times 100\%$$

Скорость (WPM) вычисляется по формуле (x – кол-во правильно набранных символов, t – время в минутах):

$$\text{WPM} = \frac{x}{5} \div t$$

В мире печати принято считать, что одно слово — это последовательность из 5 символов, включая пробелы, буквы, цифры и знаки препинания. Таким образом значение WPM определяет количество напечатанных слов в минуту.

Также в приложении реализована функция «начать заново» для повторения и закрепления результат тренировок (рис. 4).

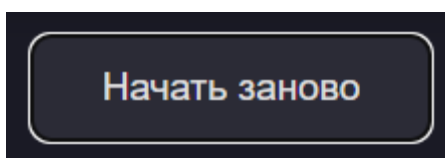


Рисунок 4. Кнопка «Начать заново»

Помимо тренировок, к приложению прилагается инструкция по слепой печати, в которой присутствуют главные правила, схема клавиатуры и положение пальцев на ней и советы для начинающих (рис. 5-6).

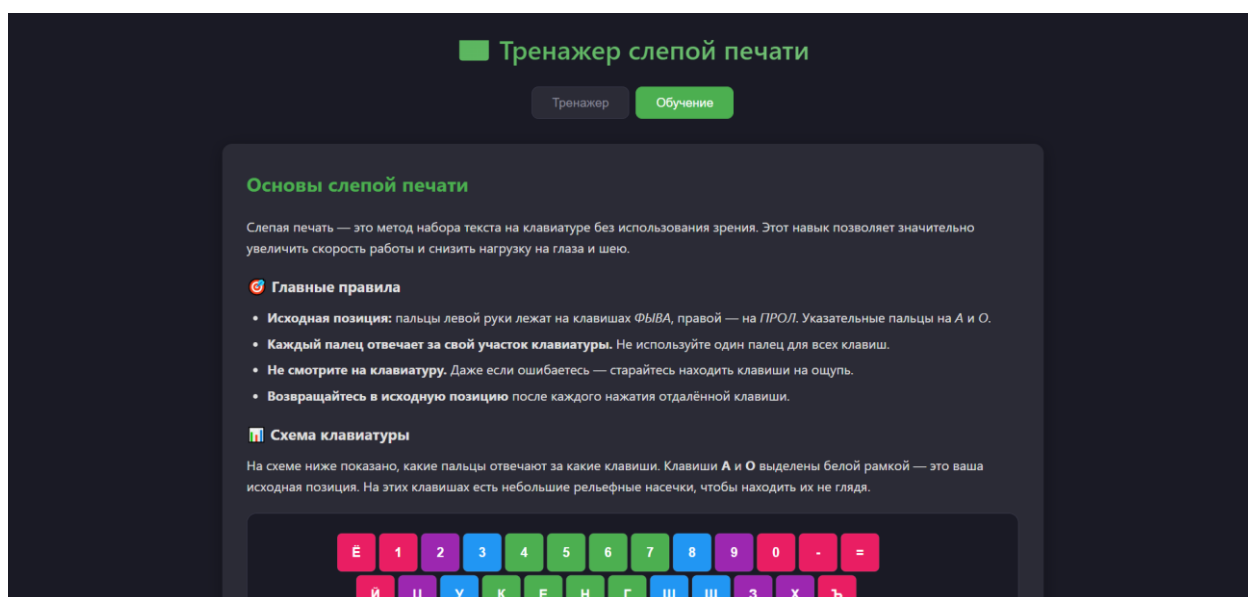


Рисунок 5. Раздел «Обучение»



Рисунок 6. Раздел «Обучение»

Анализ полученного кода показал высокую точность выполнения алгоритмических требований. Нейросеть корректно настроила перехват событий клавиатуры, обеспечив мгновенную визуальную обратную связь: система безошибочно распознает ввод, подсвечивая правильные и ошибочные символы соответствующими цветами. Математические формулы для расчета скорости набора текста и процента точности были интегрированы верно и синхронизированы с автоматическим запуском таймера. Кроме того, языковая модель успешно справилась с задачей предотвращения багов интерфейса, заблокировав стандартную прокрутку страницы при нажатии пробела.

Интеграция функции загрузки пользовательского текста продемонстрировала способность искусственного интеллекта к масштабированию программной логики. При добавлении нового текстового фрагмента приложение корректно очищает предыдущие результаты, сбрасывает таймер и обновляет активное поле, сохраняя стабильность работы всех обработчиков событий. Полученные результаты подтверждают, что структурированный промпт-инжиниринг позволяет эффективно выстраивать сложную компонентную архитектуру и управлять состояниями веб-приложения, сводя процесс разработки к грамотной постановке алгоритмических задач.

4 Вывод

Проведенное исследование подтверждает высокую эффективность автоматизированной генерации кода с помощью больших языковых моделей на примере создания интерактивного тренажера слепой печати. Использование структурированного промпт-инжиниринга и осознанный отказ от сторонних библиотек позволили спроектировать полноценное образовательное веб-приложение в рамках единого файла. Полученные результаты доказывают, что современные нейросети способны успешно справляться со сложной алгоритмической логикой, обработкой

пользовательских событий в реальном времени и динамическим управлением состояниями интерфейса.

Подобный гибридный подход наглядно иллюстрирует смену парадигмы в веб-разработке. Основной фокус внимания специалиста смещается с написания рутинного программного кода на системное проектирование и точную постановку задач для искусственного интеллекта. В перспективе описанная методология может широко применяться для оперативного прототипирования сложных цифровых продуктов, значительно снижая технический барьер и оптимизируя ресурсы на создание независимых браузерных решений.

Библиографический список

1. Рогатин С. В., Антошкин В. А. Разработка компьютерных игр по технологии html5 //Информатика и прикладная математика. 2015. №. 21. С. 12-16. URL: <https://elibrary.ru/item.asp?id=25030492>
2. Рахмонов А. Б. Вайб-кодинг как инструмент цифровой педагогики: функциональные возможности и механизмы реализации //The Lingua Spectrum. 2025. Т. 12. №. 2. С. 552-556. URL: <https://lingvospektr.uz/index.php/lngsp/article/view/1329>
3. Елманов И. В. Инженерный подход к ии: использование spec driven development для валидации vibe coding //Вестник науки. 2025. Т. 4. №. 12 (93). – С. 716-723. URL: <https://cyberleninka.ru/article/n/inzhenernyy-podhod-k-ii-ispolzovanie-spec-driven-development-dlya-validatsii-vibe-coding>
4. Мухаметов Д. Р. Политический вайб-маркетинг: возможности искусственного интеллекта для продвижения образа будущего //Известия Тульского государственного университета. Гуманитарные науки. 2025. №. 3. С. 27-33. URL: <https://cyberleninka.ru/article/n/politicheskiy-vayb-marketing-vozmozhnosti-iskusstvennogo-intellekta-dlya-prodvizheniya-obraza-buduschego>
5. Степанов А. Д., Богданов Л. А., Шишкова Д. К., Кутихин А. Г. Применение инструментов на основе искусственного интеллекта в экспериментальной и трансляционной кардиологии: обзор //Российский кардиологический журнал. 2026. Т. 31. №. S2. С. 114-122. URL: <https://cyberleninka.ru/article/n/primenenie-instrumentov-na-osnove-iskusstvennogo-intellekta-v-eksperimentalnoy-i-translyatsionnoy-kardiologii-obzor>