

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего
образования
«КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ
им. В.П. АСТАФЬЕВА»
(КГПУ им. В.П. Астафьева)

Институт/факультет

Математики, физики и информатики

(полное наименование института/факультета/филиала)

Выпускающая кафедра

Информатики и информационных технологий в
образовании

(полное наименование кафедры)

Сумбулова Софья Васильевна

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**ПРИМЕНЕНИЕ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ ДЛЯ
ИНДИВИДУАЛИЗАЦИИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ**

Направление подготовки: 44.03.05 Педагогическое образование

Направление (профиль) образовательной программы:
Математика, информатика

ДОПУСКАЮ К ЗАЩИТЕ

Зав. кафедрой ИИТО:
д-р пед. наук, профессор
Пак Н.И.

06.06.2025

(дата, подпись)

Руководитель:

канд. физ.-мат. наук, доцент каф. ИИТО
Романов Д.В.

06.06.2025

Романов

(дата, подпись)

Дата защиты «17» июня 2025 г.

Обучающийся:

Сумбулова С.В.

(дата, подпись)

Оценка отлично

Красноярск 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	2
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ИНДИВИДУАЛИЗАЦИИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ	5
1.1. Анализ содержания раздела «Алгоритмы и программирование» в школьном курсе информатики	5
1.2. Возможности и ограничения цифровых сервисов в индивидуализации образовательного процесса.....	14
Выводы по первой главе	21
ГЛАВА 2. МЕТОДИЧЕСКАЯ СИСТЕМА ИНДИВИДУАЛИЗАЦИИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ НА ОСНОВЕ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ	23
2.1. Разработка системы промтов для автоматизированного создания индивидуальных дидактических материалов	23
2.2. Методика интеграции системы в учебный процесс.....	34
Установка системы.....	34
Работа с системой.....	36
Промпт-инжиниринг	41
Выводы по второй главе	43
ЗАКЛЮЧЕНИЕ	44
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	46

ВВЕДЕНИЕ

Проблема организации всестороннего развития обучающихся с учетом их индивидуальных особенностей в современной системе обучения является одной из ключевых на сегодняшний момент. Одним из направлений в решении данного вопроса можно выделить индивидуализацию образовательного процесса - подход в образовании, при котором учебный процесс строится с учетом темпа усвоения материала, способностей и интересов каждого обучающегося.

Учет темпа работы и возможностей обучающихся является основой в обучении программированию. В процессе изучения тематического раздела «Алгоритмы и программирование» обучающиеся с разным уровнем подготовки сталкиваются с существенными трудностями: одни не успевают освоить базовые концепции, другие теряют мотивацию из-за недостаточной сложности заданий.

Современное образование столкнулось с новыми вызовами в связи с появлением доступных больших языковых моделей, которые позволяют современным школьникам мгновенно получать готовые решения задач не прикладывая значительных усилий и даже не осознавая сути самой задачи. Это привело к обесцениванию самостоятельной работы, снижению глубины понимания материала и росту академической неуспеваемости. Однако вместо того, чтобы бороться с нецелевым использованием искусственного интеллекта, разумнее интегрировать его в учебный процесс, например, использовать его как инструмент индивидуализации обучения.

С помощью искусственного интеллекта мы можем автоматизировать индивидуализацию обучения программированию, что позволит уменьшить негативные последствия бездумного использования искусственного интеллекта школьниками, превратив его в инструмент выстраивания

процесса осознанного обучения, повысить качество учебного процесса за счет индивидуальных дидактических материалов.

Противоречия: рост эффективности обучения программированию прослеживается при адаптации заданий под уровень знаний и темп работы каждого обучающегося, в то время как доступные учителю цифровые средства пока не могут автоматически подстраиваться под индивидуальные потребности каждого школьника.

Проблема: как обеспечить эффективную индивидуализацию обучения программированию?

Цель: обосновать и разработать методику применения больших языковых моделей для автоматизированной индивидуализации обучения программированию.

Объект: процесс обучения программированию в школьном курсе информатики.

Предмет: автоматизация процесса создания индивидуальных дидактических материалов с помощью больших языковых моделей.

Задачи:

1. Проанализировать содержание тематического раздела «Алгоритмы и программирование» школьного курса информатики и выявить проблемы, возникающие у обучающихся в ходе изучения данного раздела.
2. Оценить возможности и ограничения цифровых сервисов в контексте индивидуализации образовательного процесса.
3. Разработать систему промтов для автоматизированной индивидуализации обучения программированию.
4. Разработать методические рекомендации для учителя по интеграции данной системы в учебный процесс.

Выпускная квалификационная работа включает в себя введение, две главы, заключение, библиографический список, состоящий из 31 источника. Общий объем работы составляет 50 страниц.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ИНДИВИДУАЛИЗАЦИИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ

1.1. Анализ содержания раздела «Алгоритмы и программирование» в школьном курсе информатики

В основной школе, а именно с 7 по 9 классы, информатика может изучаться в двух форматах: базовый и углубленный уровни. Подробная информация о данных уровнях изучения представлена в таблице 1.

Таблица 1

Анализ уровней изучения школьного курса информатики

Школьный курс информатики		
Характеристика	Базовый уровень	Углубленный уровень
Классы	с 7 по 9 классы	с 7 по 9 классы
Часы в неделю	1 час	2 часа
Общее кол-во часов	102 часа	204 часа
Основные темы	<p>7 класс:</p> <ul style="list-style-type: none"> • информация и информационные процессы; • представление информации; • компьютер — универсальное устройство обработки данных; • программы и данные; • компьютерные сети; • текстовые документы; • компьютерная графика; • мультимедийные презентации. <p>8 класс:</p> <ul style="list-style-type: none"> • системы счисления; • элементы математической логики; • исполнители и алгоритмы, алгоритмические конструкции; • язык программирования; • анализ алгоритмов. 	<p>7 класс:</p> <ul style="list-style-type: none"> • компьютер – универсальное устройство обработки данных; • программы и данные; • компьютерные сети; • информация и информационные процессы; • представление информации; • исполнители и алгоритмы, алгоритмические конструкции; • компьютерная графика и анимация; • текстовые документы; • компьютерная графика; • мультимедийные презентации. <p>8 класс:</p> <ul style="list-style-type: none"> • системы счисления; • элементы математической логики;

	<p>9 класс:</p> <ul style="list-style-type: none"> • разработка алгоритмов и программ; • управление; • моделирование как метод познания; • электронные таблицы; • глобальная сеть Интернет и стратегии безопасного поведения в ней; • работа в информационном пространстве; • информационно-коммуникационные технологии в современном обществе. 	<ul style="list-style-type: none"> • язык программирования; • электронные таблицы. <p>9 класс:</p> <ul style="list-style-type: none"> • глобальная сеть Интернет и стратегии безопасного поведения в ней; • работа в информационном пространстве; • моделирование как метод познания; • разработка алгоритмов и программ; • управление; • электронные таблицы; • информационные технологии в современном обществе.
--	---	--

Основным разделом, изучаемым в каждом из представленных уровней, является «Алгоритмы и программирование». В соответствии с примерной рабочей программой Л.Л. Босовой, на изучение данного раздела на базовом уровне отводится 29 часов: 21 час в 8 классе и 8 часов в 9 классе [12]. На углубленном уровне отводится 86 часов: 24 часа в 7 классе, 34 часа в 8 классе и 28 часов в 9 классе [23]. При изучении раздела рассматриваются следующие темы: «Исполнители и алгоритмы. Алгоритмические конструкции», «Язык программирования», «Анализ алгоритмов», «Разработка алгоритмов и программ», «Управление». Поурочное планирование по данным темам раздела представлено в таблице 2.

Таблица 2

Поурочное планирование по тематическому разделу «Алгоритмы и программирование» (базовый уровень)

Тема	Кол-во часов
Тематический раздел «Алгоритмы и программирование»	
8 класс	
Понятие алгоритма. Исполнители алгоритмов	1
Свойства алгоритма. Способы записи алгоритма	1

Алгоритмическая конструкция «следование». Линейный алгоритм	1
Алгоритмическая конструкция «ветвление»: полная и неполная формы	1
Алгоритмическая конструкция «повторение»	1
Формальное исполнение алгоритма	1
Разработка несложных алгоритмов с использованием циклов для управления формальными исполнителями	1
Разработка несложных алгоритмов с использованием циклов и ветвлений для управления формальными исполнителями	1
Выполнение алгоритмов	1
Обобщение и систематизация знаний. Контрольная работа по теме «Исполнители и алгоритмы. Алгоритмические конструкции»	1
Язык программирования. Система программирования	1
Переменные. Оператор присваивания	1
Программирование линейных алгоритмов	1
Разработка программ, содержащих оператор ветвления	1
Диалоговая отладка программ	1
Цикл с условием	1
Цикл с переменной	1
Обработка символьных данных	1
Обобщение и систематизация знаний по теме «Язык программирования»	1
Анализ алгоритмов. Определение возможных результатов работы алгоритма при заданном множестве входных данных	1
Анализ алгоритмов. Определение возможных входных данных, приводящих к данному результату	1
Итого	21
9 класс	
Разбиение задачи на подзадачи. Составление алгоритмов и программ с использованием ветвлений, циклов и вспомогательных алгоритмов	1
Одномерные массивы	1
Типовые алгоритмы обработки массивов	1
Сортировка массива	1
Обработка потока данных	1
Обобщение и систематизация знаний. Контрольная работа по теме «Разработка алгоритмов и программ»	1
Управление. Сигнал. Обратная связь	1

Роботизированные системы	1
Итого	8

В результате освоения основ алгоритмизации и программирования в рамках школьной программы основного общего образования обучающиеся должны научиться:

- объяснять смысл ключевых понятий данного раздела;
- анализировать последовательности команд на наличие у них свойств алгоритма: дискретность, детерминированность, понятность, результативность, массовость;
- по заданной блок-схеме определять, для решения какой задачи предназначен данный алгоритм;
- анализировать изменение значений величин при пошаговом выполнении алгоритма;
- на основе выбранного метода решения задачи определять, какие алгоритмические конструкции могут войти в алгоритм;
- сравнивать различные алгоритмы решения одной задачи;
- создавать, выполнять вручную и на компьютере несложные алгоритмы с применением циклических и условных конструкций для управления исполнителями, такими как «Робот», «Черепашка», «Чертежник»;
- выполнять готовые алгоритмы при конкретных входных данных;
- строить для исполнителя арифметических действий цепочки команд, дающих требуемый результат при конкретных исходных данных;
- программировать линейные алгоритмы, предполагающие вычисление арифметических, строковых и логических выражений;
- разрабатывать программы, содержащие оператор ветвления и оператор цикла;
- анализировать готовые алгоритмы и программы;

- осуществлять разбиение исходной задачи на подзадачи;
- разрабатывать программы, включающие в себя подпрограмму [12].

На основе анализа методических пособий можно сделать вывод, что изучение раздела «Алгоритмы и программирование» в рамках школьного курса информатики начинается с работы в системе программирования КуМир. Данная система имеет ряд преимуществ: русский язык программирования, визуализация процесса выполнения программы и простой интерфейс (рисунок 1.1.), что позволяет обучающимся быстро освоить начала алгоритмизации и программирования [18].

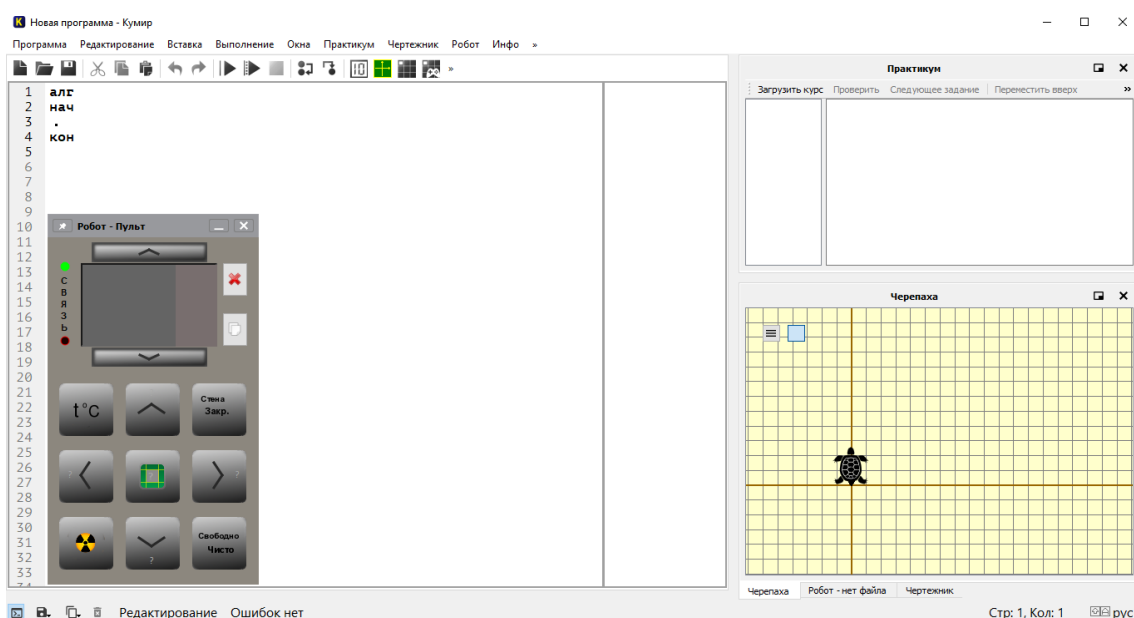


Рисунок 1.1. Система программирования КуМир

В ходе дальнейшего изучения данного раздела осуществляется переход от системы КуМир к освоению более сложного универсального языка программирования. Выбор конкретного языка зависит от целей и задач обучения, а также от уровня подготовки обучающихся. В рамках школьного курса информатики может изучаться один из следующих языков программирования: Python, C++, Паскаль, Java, C#, Школьный Алгоритмический Язык [12].

В современной школе большинство учителей отдают предпочтение языку Python благодаря ряду его преимуществ перед другими языками программирования. Python отличается наиболее простым синтаксисом, который легко усваивается обучающимися, что позволяет сосредоточиться на изучении основ программирования, не тратя время на сложные конструкции и особенности языка. Кроме того, программы на Python легче и быстрее создавать, что является полезным при подготовке и сдаче единого государственного экзамена [26]. Основываясь на данных индекса ТЮВЕ, представленных на рисунке 1.2., Python является одним из самых популярных языков программирования в мире [7]. Им активно пользуются в веб-разработке, анализе данных, машинном обучении и тестировании, что открывает карьерные перспективы перед обучающимися, готовя их к работе в стремительно развивающихся областях информационных технологий.











Май 2025 года	Май 2024 года	Изменение	Язык программирования	Рейтинги	Изменение
1	1		 Python	25.35%	+9.02%
2	3	▲	 C++	9.94%	+0.41%
3	2	▼	 C	9.71%	-0.27%
4	4		 Java	9.31%	+0.62%
5	5		 C#	4.22%	-2.27%
6	6		 JavaScript	3.68%	+0.66%
7	8	▲	 Вперед	2.70%	+1.10%
8	7	▼	 Visual Basic (Базовый)	2.62%	+0.61%
9	11	▲	 Delphi/Объектный Паскаль	2.29%	+1.05%
10	9	▼	 SQL	1.90%	+0.45%
11	10	▼	 Фортран	1.78%	+0.53%
12	24	▲	 R	1.46%	+0.71%

Рисунок 1.2. Рейтинг языков программирования

Но даже не смотря на ряд достоинств языка Python, в ходе обучения программированию учителя сталкиваются с рядом проблем. Рассмотрим некоторые из них: отсутствие единой методической базы, недостаточное количество часов, выделенных на изучение, быстрая потеря мотивации,

разный темп усвоения материала обучающимися и технические трудности, замедляющие темп работы.

На основе анализа учебников, одобренных министерством просвещения, можно сделать вывод об отсутствии единой методической базы по изучению тематического раздела «Алгоритмы и программирование». В учебнике для 8 класса Л.Л. Босовой основы программирования начинаются с изучения языка Pascal, по учебнику К.Ю. Полякова программирование также изучается на языке Pascal, но начинается оно с 7 класса. Изучение языков программирования по учебнику И.Г. Семакина начинается только в 9 классе. В связи с отсутствием единой методической базы обучения программированию, определение форм и методов представления материала выполняется непосредственно самим учителем. Из-за этого возрастает нагрузка, справиться с которой могут далеко не все педагоги. В результате одни учителя представляют материал в творческом формате, используя интерактивы, креативные подходы, тем самым формируют и удерживают интерес обучающихся, в то время как другие учителя ограничиваются простым пересказом материала учебника [32]. Это приводит к значительному различию качества образования, одни обучающиеся приобретают прочные знания и умения, а другие напротив, зачастую испытывают трудности в понимании и применении материала.

В основной школе на изучение алгоритмов и программирования отводится от 15 до 27,5% учебного времени по информатике. Сравнивая данные показатели и планируемые результаты по освоению рассматриваемой нами темы, появляется сомнение в реальности выполнения намеченного плана [14].

Следующей трудностью в обучении программированию является быстрая потеря мотивации у обучающихся, она может быть связана со следующими факторами: сложность программирования, отсутствие четкого осознания важности изучаемого материала, выполнение однообразных действий [10]. Мотивация является ключевым фактором эффективного

обучения, вследствие ее снижения может прослеживаться ухудшение усвоения материала, формирование негативного отношения к учебному процессу.

Раздел «Алгоритмы и программирование» считается одним из самых сложных в курсе школьной информатики в силу абстракции и технологической насыщенности. Поэтому в процессе обучения темп восприятия и усвоения у разных учеников может различаться: одни остаются позади, испытывая трудности, другие справляются слишком легко, утрачивая интерес, а для остальных данный темп работы является оптимальным [11].

Обучение программированию часто сопровождается возникновением технических трудностей, проблем, возникающих на этапе написания кода программы и ее запуска. Рассмотрим технические трудности, с которыми сталкиваются обучающиеся как на начальном этапе изучения программирования, так и на этапе систематизации полученных знаний. На начальном этапе изучения программирования обучающиеся знакомятся с базовыми понятиями и их применением. На этом этапе многие важные идеи вызывают трудности или остаются непонятными, такие как:

- принцип последовательности выполнения команд;
- отличие программы от математического уравнения;
- состояния исполнителя на указанный момент;
- области видимости переменных;
- ветвления, вложения, дерево условий;
- мысленная изоляция вложенного цикла от внешнего;
- логика перевода задачи в алгоритм ее решения;
- выделение повторяющегося поведения в отдельные модули;
- имена переменных и типов данных не означают для компилятора ничего и не гарантируют появление соответствующих свойств и операторов;
- области видимости: имена полей, имена структур, имена массивов, локальные и глобальные переменные, аргументы;

- работа с динамическими адресами данных.

На следующем этапе, этапе усвоения и автоматизации, обучающиеся сталкиваются с новыми трудностями, они не понимают:

- как проектировать сложные программы;
- что такое алгоритмическая сложность программы;
- как пользоваться сложными структурами данных;
- какую следует использовать библиотеку для выполнения задачи;
- необходимость использования библиотеки;
- каким образом на канве будут отображаться графики;
- как правильно задать координаты точек графика;
- как идет прорисовка изображения в целом;
- каким образом обрабатываются данные в графических структурах;
- как применить описанную структуру при решении задачи.

Решение рассмотренных проблем ложится на плечи учителя, целью которого является поиск педагогических технологий, которые будут поддерживать и мотивировать обучающихся на протяжении всего образовательного процесса. Среди подобных технологий можно выделить индивидуализацию обучения, ключевой целью которой является учет индивидуальных особенностей обучающихся при организации учебного процесса.

1.2. Возможности и ограничения цифровых сервисов в индивидуализации образовательного процесса

Понятие «индивидуализация обучения» рассматривается во многих источниках, но формулировки его определения разнятся.

Данное понятие используется в статье О.И. Вагановой, Е.С. Павловой, О.Г. Шагаловой, И.Р. Ворониной, где «индивидуализация обучения» рассматривается как «совокупность психологических, организационно-управленческих, педагогических, учебно-методических приемов, методов и технологий, которые обеспечивают индивидуальный подход к каждому обучающемуся» [15].

В свою очередь С.М. Авдеева, Л.Л. Босова, Н.В. Никуличева, С.С. Хапаева определяют под индивидуализацией обучения организацию образовательного процесса по программам общего и дополнительного образования, предусматривающую обязательный учет индивидуальных особенностей каждого обучающегося. Ключевая цель данной технологии - обеспечение оптимальных условий для полного раскрытия и реализации потенциала каждого ученика, включая формирование навыков самостоятельного управления своей образовательной траекторией [8].

Данное понятие рассматривается в работе Г.Х. Воистинова, Е.А. Чернова, М.Р. Байназарова, под которым понимается отдельный метод обучения, содержание, технология и темп обучения в котором подстроены под способности и интересы каждого обучающегося [16].

Рассмотрев данные определения понятия «индивидуализация обучения» можно выявить общие черты, а именно, под индивидуализацией обучения мы будем понимать технологию организации учебного процесса, целью которой является адаптация содержания, методов и темпа деятельности обучающегося под его индивидуальные особенности.

Данная технология позволяет осуществлять наблюдение за каждым действием обучающегося при решении конкретной задачи, анализировать

прогресс от открытия нового знания к формированию умений и навыков, выявлять возникающие трудности и успехи, на основе чего вовремя вносить корректировки в деятельность учителя и обучающегося.

Условия, которые позволяют оценить успешную реализацию технологии индивидуализации обучения, делятся на три группы: процессуальная, содержательная и управленческая [19].

Процессуальная группа включает в себя требования к темпу и формам обучения, к организации взаимодействия на занятии:

- персональный темп обучения;
- завершенность каждого этапа учебного процесса, перед переходом к новому материалу, необходимо полностью освоить предыдущий;
- многообразие форм организации учебной деятельности на уроке;
- активное сотрудничество между обучающимися.

Содержательная группа состоит из условий, определяющих содержание и результат учебного действия, к ним относятся:

- развитие метапредметных умений;
- разнообразие образовательных ситуаций, форм взаимоотношений учащихся;
- наличие дидактических материалов разного уровня по одной и той же теме.

Управленческие условия описывают требования к деятельности учителя в ходе образовательного процесса:

- включение обучающихся в совместный процесс рефлексии;
- диагностика учебных задач;
- владение информацией о каждом обучающемся и умение применить ее для осуществления индивидуализации обучения [19].

К организационно-педагогическим методам индивидуализации обучения относятся подбор индивидуальных заданий, разработка индивидуального плана урока и проведение индивидуальной работы с

обучающимся. Основная идея выбранных методов представлена на рисунке 1.3.

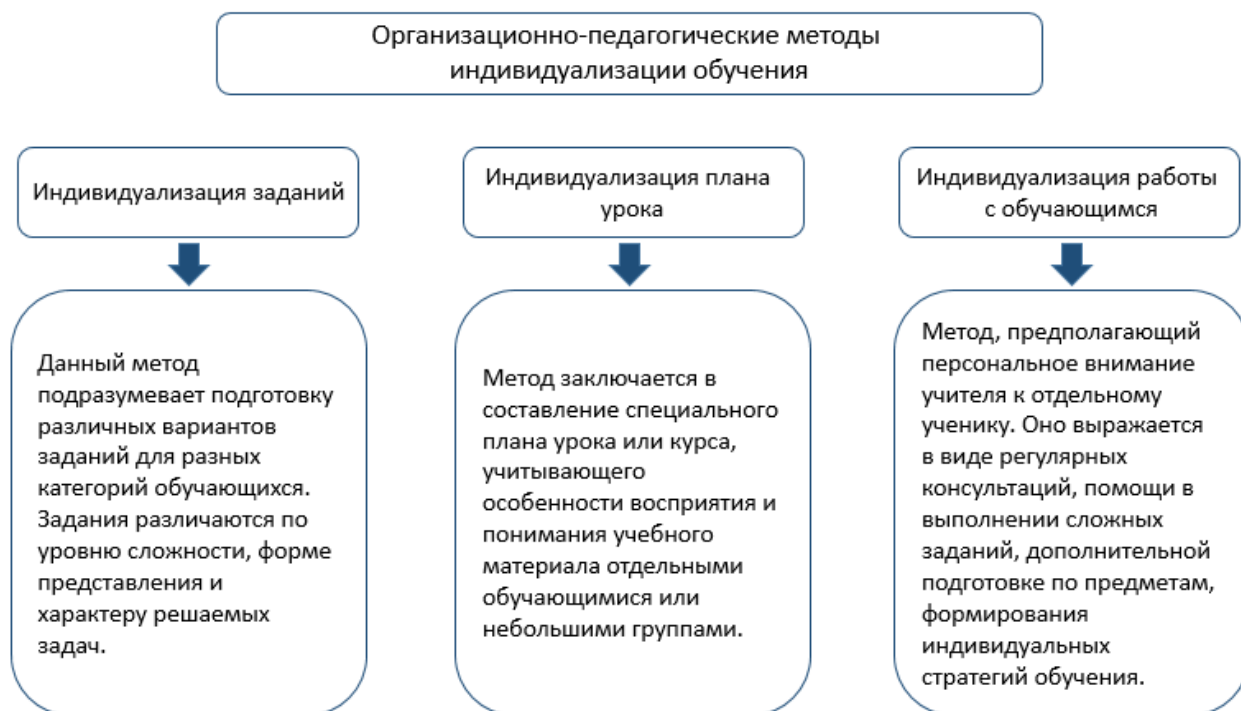



Рисунок 1.3. Характеристика методов индивидуализации обучения

В ходе обучения программированию целесообразно воспользоваться методом индивидуализации заданий, в силу того, что программирование характеризуется как область, в которой обучающиеся развиваются скачкообразно, быстро осваивая один материал и ощущая трудности в другом. Поэтому важно предлагать задачи различного уровня сложности и в разных формулировках, соответствующие возможностям каждого ученика. Также применение метода индивидуализации заданий проще всего автоматизируется. Преподаватель может заранее подготовить наборы заданий для разных уровней подготовки, предоставив возможность учащимся выбирать подходящий вариант. Затем останется лишь проверять выполненные задания и помогать обучающимся в решении возникающих вопросов. При решении каждой задачи, обучающийся сразу может увидеть результат своей работы и осознать собственный успех или неудачу, что стимулирует на дальнейшее продвижение вперед.

Использование цифровых технологий позволяет эффективно организовать процесс обучения с учетом индивидуальных особенностей каждого обучающегося, а именно темпа восприятия и усвоения материала. Рассмотрим ряд цифровых сервисов, позволяющих выполнить данную задачу.

Plario (рисунок 1.4.). Первая в России онлайн система аддитивного обучения. В данное время платформа ориентирована только на обучение математике старшеклассников и студентов, с целью закрытия пробелов в школьном курсе математики. Plario создана как онлайн-платформа «цифрового репетитора», обеспечивающая индивидуализацию обучения. К преимуществам данной платформы относятся: модуль управления контентом, то есть преподаватель может сам создавать и редактировать материалы обучения; на основе входного тестирования создается цифровой двойник студента, для которого формируется индивидуальный учебный план; возможность интеграции с системами управления обучением (LMS). Но также можно выделить и минусы Plario: доступ к данной системе осуществляется только через официальную заявку ВУЗа или школы; узконаправленность, обучение только по математике [5].



Цифровизация

Вход в PLARIO

Связаться с нами

Адаптивный подбор упражнений

Система индивидуально подбирает траекторию обучения в зависимости от уровня подготовки и прогресса студента

Контроль знаний в режиме реального времени

В процессе выполнения заданий система оценивает прогресс обучающегося и корректирует траекторию обучения

Высокая вовлечённость и мотивация

Элемент геймификации подтвердил свою эффективность для освоения учебного материала

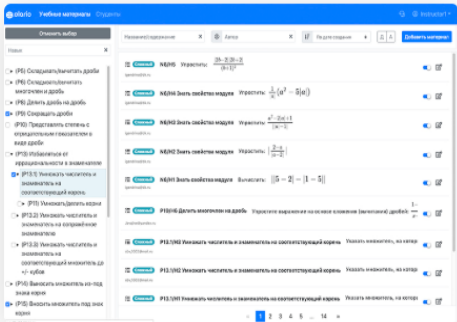


Рисунок 1.4. Plario - онлайн-система аддитивного обучения

Duolingo (рисунок 1.5.). Платформа для бесплатного изучения языка. На основе работы искусственного интеллекта сложность и темп обучения подбираются под каждого пользователя индивидуально. Учитывая допущенные ошибки, система формирует контент, который позволяет отработать и закрепить материал, с которым возникли трудности. Но платформа осуществляет обучение только языкам, нет возможности редактировать и отслеживать образовательную траекторию учителем [1].

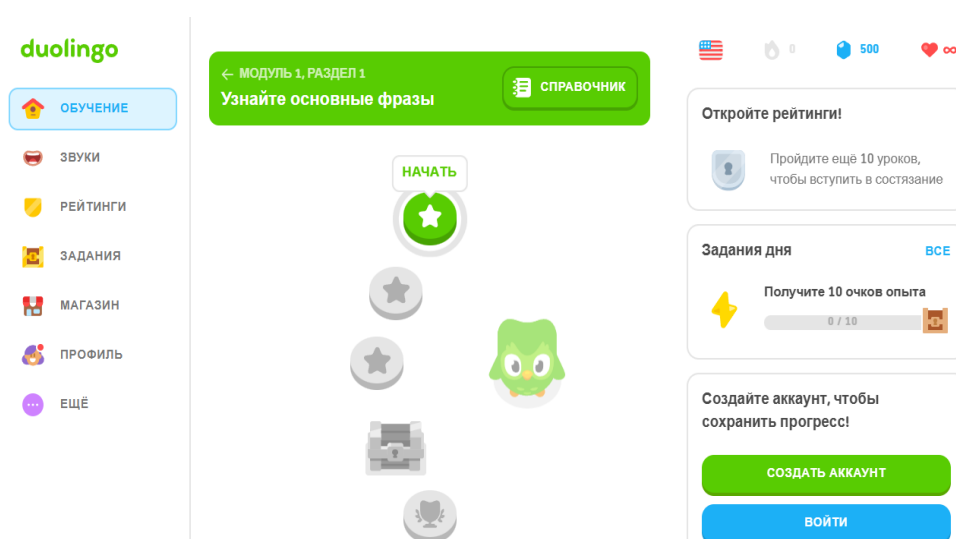


Рисунок 1.5. Duolingo - платформа для изучения языка

Mr. Ranedeer - AI Tutor. Онлайн-репетитор, который использует технологию GPT-4. Уникальность данного ресурса заключается в способности использования искусственного интеллекта для создания индивидуальных курсов, которые полностью подстраиваются под потребности каждого пользователя. К основным особенностям Mr. Ranedeer относятся:

- индивидуальные курсы, осуществляется адаптация обучения под уровень знаний и цели обучающегося;
- настройка сложности, можно выбрать уровень сложности материала, от начальной школы до докторской степени;

- стили обучения: визуальный, вербальный, активный, интуитивный, рефлексивный, глобальный;
- тональность обучения: ободряющий, нейтральный, информативный, дружелюбный, юмористический;
- работа на языке, с которым может работать GPT-4.

Работа осуществляется непосредственно в самом ChatGPT, эффективнее работать в подписке Plus, но если нет такой возможности, то можно обойтись четвертой версией чата. Для запуска необходимо скопировать готовый промт в чат, после чего можно приступить к настройке и работе с онлайн-репетитором. Mr. Ranedeer работает за счет команд:

- /test: запросить тест для оценки знаний и понимания;
- /config: обновить конфигурацию/предпочтения AI Tutor;
- /plan: создать план урока;
- /start: запустить план урока;
- /continue: продолжить вывод, если он был прерван;
- /language: изменить язык AI Tutor [3].

Хотя Mr. Ranedeer - AI Tutor имеет много полезных функций, которые позволят осуществить индивидуализацию, можно выделить и ограничения, не позволяющие использовать данный ресурс в процессе обучения. К ним относятся: учитель не сможет контролировать образовательный контент; работа выполняется в чате, промт будет виден обучающимся, что недопустимо в силу приватности информации; существует ограничения по использованию иностранных платформ в образовательном процессе.

ИИ Препад (рисунок 1.6.) - репетитор на базе искусственного интеллекта, адаптируемый к знаниям и интересам пользователя. Процесс обучения организован в виде общения с искусственным интеллектом в формате диалога (рисунок 1.7.) [17].



Рисунок 1.6. - ИИ Преподаватель - онлайн-образовательная платформа

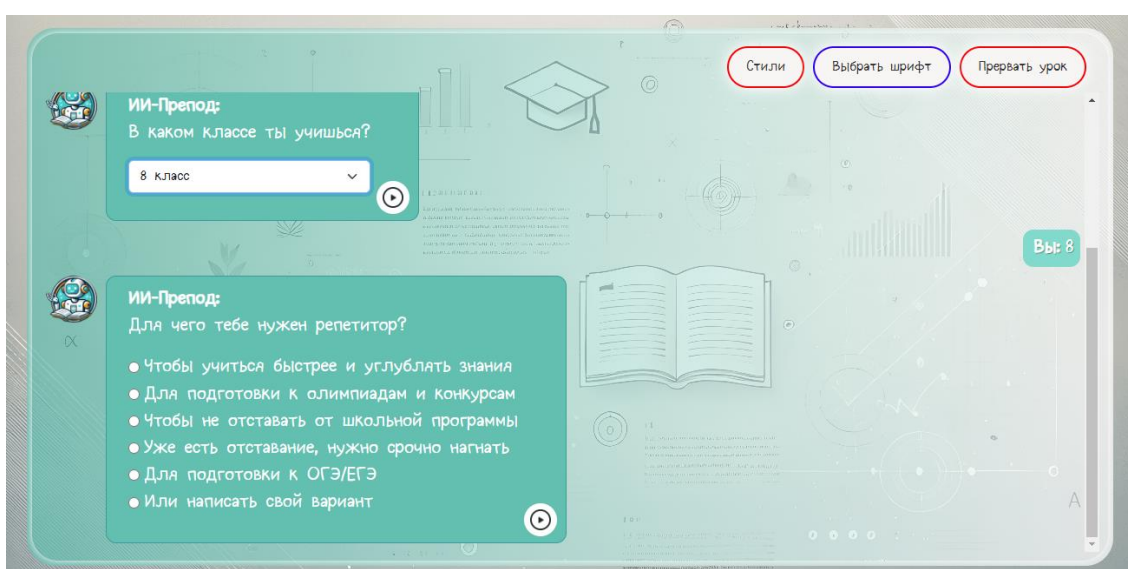


Рисунок 1.7. - Чат с ИИ

К основным преимуществам данной платформы можно отнести: персонализация обучения; доступность по времени; контроль прогресса; объективность оценки.

Но платформа не позволяет контролировать процесс обучения учителю и обучение на данной платформе является платным.

Учитывая анализ имеющихся ресурсов, мы можем сформулировать требования для сервиса, с помощью которого будет возможна организация индивидуализации обучения в школе:

- осуществление адаптации образовательного контента под уровень знаний, темп работы и индивидуальные особенности каждого обучающегося;
- возможность создавать учебные материалы и контролировать процесс работы непосредственно самим учителем;
- наглядное представление информации, простой интерфейс;
- результат работы должен быть виден обучающемуся сразу;
- доступность сервиса.

Сервис, позволяющий организовать индивидуализацию обучения в школе, должен использовать технологию искусственного интеллекта (ИИ). Она позволит автоматизировать создание учебных дидактических материалов, за счет чего существенно снизится нагрузка на учителя.

Выводы по первой главе

Тематический раздел «Алгоритмы и программирование» является сложным элементом всего школьного курса информатики. Данный раздел охватывает большой спектр теоретических и практических знаний, начиная с изучения элементарных алгоритмических конструкций и заканчивая созданием программ на высокоуровневых языках программирования (например, Python, Pascal, C++).

В первой главе были рассмотрены основные трудности, возникающие у учителей в ходе обучения программированию: отсутствие единой методической базы, недостаточное количество часов, выделенных на изучение, быстрая потеря мотивации, разный темп усвоения материала обучающимися и технические трудности, замедляющие темп работы.

Для решения данных проблем была выбрана технология индивидуализации обучения, так как за счет учета индивидуальных особенностей каждого обучающегося, можно удерживать мотивацию, выстраивать материал под темп усвоения, рационально использовать учебные часы и создать методическую базу, которая в последующем облегчит работу учителя.

Цифровые сервисы являются эффективным средством для индивидуализации обучения программированию. В результате проведенного анализа платформ, позволяющих адаптировать учебный контент под уровень знаний и темп работы, были сформированы требования, которыми должен обладать цифровой сервис. К ним относятся: осуществление адаптации образовательного контента под уровень знаний, темп работы и индивидуальные особенности каждого обучающегося; возможность создавать учебные материалы и контролировать процесс работы непосредственно самим учителем; наглядное представление информации, простой интерфейс; результат работы должен быть виден обучающемуся сразу; доступность сервиса.

Использование искусственного интеллекта в работе цифрового сервиса позволит организовать автоматизацию создания учебного контента с учетом индивидуальных особенностей каждого обучающегося.

ГЛАВА 2. МЕТОДИЧЕСКАЯ СИСТЕМА ИНДИВИДУАЛИЗАЦИИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ НА ОСНОВЕ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

2.1. Разработка системы промтов для автоматизированного создания индивидуальных дидактических материалов

Генеративный искусственный интеллект (ИИ) - это технология искусственного интеллекта, которая на основе запроса в диалоговом окне создает контент (текст, изображение, видео, музыка, программный код). Обучение генеративного ИИ осуществляется на основе больших массивов данных, полученных из интернета: содержимое веб-страниц, книг, статей, форумов и т.д..

Генеративный ИИ основан на технологиях машинного обучения, работа которых автоматически улучшается по мере поступления новых данных. Одним из типов машинного обучения, являются искусственные нейронные сети, работающие аналогично структуре и функционированию человеческого мозга. Нейронные сети также подразделяются на типы, к которым относятся технологии генерации текста и изображений (таблица 3) [26].

Таблица 3

Методы, используемые в генеративном искусственном интеллекте

Машинное обучение (МО)		Тип ИИ, который использует данные для автоматического улучшения своей производительности
Искусственная нейронная сеть (ИНС)		Тип машинного обучения, аналогичный структуре и функционированию человеческого мозга (например, синаптические связи между нейронами)
Текстовый генеративный ИИ	Трансформеры общего назначения	Тип ИНС, способный концентрироваться на различных частях данных, чтобы определить, как они связаны друг с другом

	Большие языковые модели (БЯМ)	Тип преобразователя общего назначения, который обучается на огромных объемах текстовых данных
	Генеративный предварительно обученный трансформер (GPT)	Тип большой языковой модели, предварительно обученной на обширных данных, повышающих ее способность к восприятию языковых нюансов и созданию качественного текста, адаптированного к конкретному контексту
Генератор изображений ИИ	Генеративно-состязательные сети	Типы нейронных сетей, используемых для генерации изображений
	Вариационные автокодировщики	

Системы ИИ для генерации текста называют большими языковыми моделями (БЯМ), они выполняют обработку естественного языка и генерацию человекоподобных ответов. Обучение языковой модели включает два этапа: предварительное обучение (проходит на большом массиве данных, выполняется метод обучения без учителя); точная настройка (к конкретной задаче применяется модель, прошедшая предварительное обучение, далее корректируются параметры с помощью обучения с учителем) [24]. Самой популярной БЯМ считается ChatGPT, разработанный компанией OpenAI в 2018 году. На данный момент насчитывается четыре версии чата, каждая версия отличается увеличением числа параметров, повышением качества ответов и способностью справляться с задачами более сложного уровня [4].

Конечно, БЯМ не сводятся к одному ChatGPT, их количество постоянно растет, так же как и конкуренция в этой области. Среди отечественных разработок также имеются достойные БЯМ, отлично справляющиеся со своей задачей. GigaChat - нейросеть на русском языке от Сбера, также работает в формате чата. Помимо генерации текста, обладает опцией создания изображений. История диалогов с моделью сохраняется, вход в систему выполняется через учетную запись Сбербанк Онлайн [2].

Использование БЯМ может показаться простым процессом, достаточно ввести запрос и мгновенно получить ответ. Но и в этом процессе есть правила, которые помогают оптимизировать работу модели и получить именно тот результат, который хочет пользователь [25]. Prompt engineering - разработка и оптимизация запросов (промтов) с целью эффективного использования БЯМ [27]. Основные методы по созданию эффективных запросов (промтов):

1. Наличие четкой инструкции. Формулируя промт, следует уточнять детали желаемого ответа, попросить модель принять ту или иную роль, указать шаги, необходимые для выполнения запроса, привести примеры и установить желаемый объем ответа.

Пример:

\\Объясни понятие языка программирования новичкам в IT сфере.

Ты опытный учитель информатики, объясняющий материал доступно и понятно.

План ответа:

1. Дать общее определение языка программирования.
2. Описать, зачем нужны языки программирования и какие задачи они решают.
3. Привести конкретные примеры языков программирования и рассказать, где они используются.
4. Сделать простое сравнение двух категорий языков программирования.

Желаемая длина текста: 700-800 символов.\\

2. Использование справочного текста. Ответ модели строится на основе текстового источника, который включается в запрос (файл, ссылка или скопированный текст).

Пример:

\\Кратко расскажи о значении информатики в современном мире, основываясь на приведенном тексте.\\

3. Разбиение сложных заданий на подзадачи. В большом диалоге с моделью следует резюмировать промежуточный результат.

Пример:

\\Сделай краткое резюме нашего предыдущего обсуждения организации конференции, охватывая основные темы и принятые решения. Объем резюме – около 500 символов.\\

4. Использование запроса «цепочка мысли». Данный метод поможет модели эффективно рассуждать и найти более надежное решение [6].

Пример:

\\Реши уравнение $x^2 + 3x - 10 = 0$, показав все шаги вычислений.\\

Используя указанные приемы, сформируем систему промтов, которая позволит автоматизировать создание индивидуальных дидактических материалов по тематическому разделу «Алгоритмы и программирование». Осуществлять запросы будем в GigaChat.

Система содержит следующие элементы: формулировка, описание и результат работы промта. Данные, выделенные курсивом, являются настраиваемыми параметрами, которые учитель может менять (например, тема урока, данные об обучающихся, формулировки задач и т.д.).

1. Промт:

\\Ты опытный учитель информатики, проводящий урок по теме «Цикл с условием». Для начала работы важно установить контакт с учеником. Напиши приветствие и попроси ученика представиться. Используй следующую форму обращения: *Привет! Напиши своё имя.*\\

Данный промт необходим для установления контакта между системой и обучающимся перед началом работы. Приветствие помогает снять напряжение и настроить ученика на продуктивное общение. Благодаря запросу имени, система анализирует данные, которые предоставил учитель об обучающемся, на основе них формирует индивидуальный образовательный контент (рисунок 2.1.).

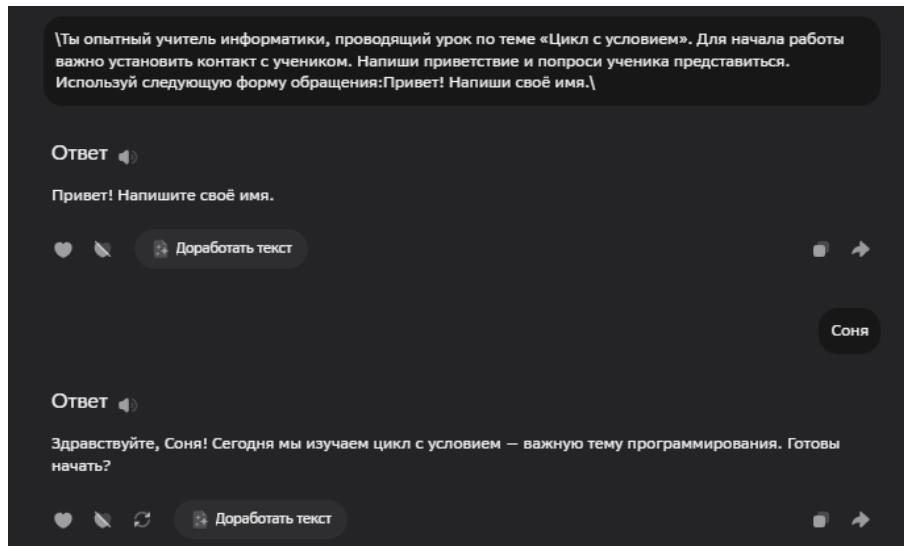


Рисунок 2.1. - Промт №1

2. Промт:

\\Начни работу по данной теме. Для этого на выбор дай обучающемуся 2 задачи.

Задай вопрос следующим образом: *«С какой задачи ты хочешь начать?»*

1. Найти сумму всех чисел от 1 до 10 с помощью цикла while.

2. Вывести первые 10 натуральных чисел с помощью цикла while.»

Запроси выбор задачи и предоставь соответствующее задание для решения.\\

Цель промта - дать возможность обучающемуся самостоятельно выбрать задачу для решения, это позволит повысить

заинтересованность и вовлеченность ученика. На основе решений выбранной задачи система далее сможет собирать данные о способностях обучающегося и уровне его освоения материала, что поможет выстроить дальнейшую работу (рисунок 2.2.).

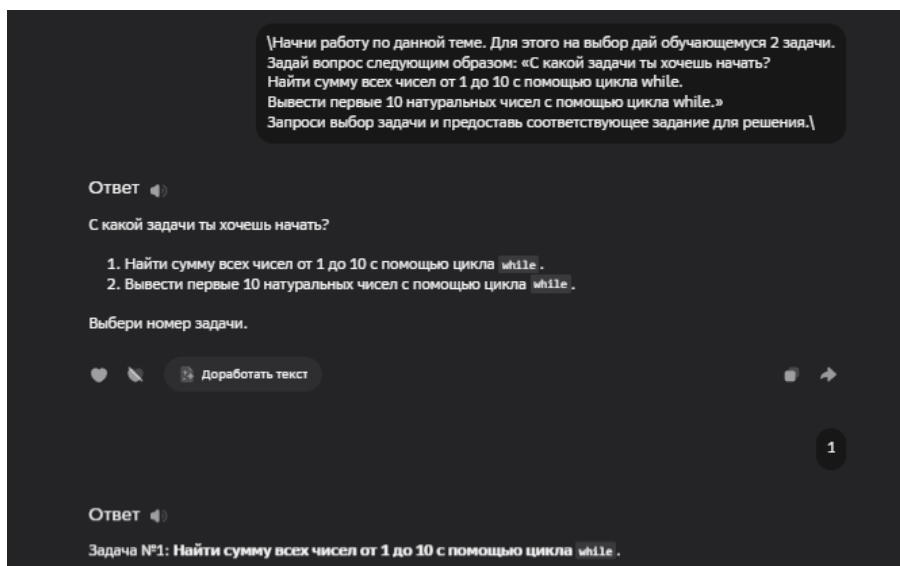


Рисунок 2.2. - Промт №2

3. Промт:

\\Поддерживай двустороннее общение, позволяя ученику задавать вопросы по возникшим затруднениям.

Например, можно сказать: «Если у тебя возникли вопросы по задаче или её решению, можешь задать их мне.»

Однако не давай готовый ответ, дай ученику подумать самостоятельно. Задавай вопросы, которые наведут на правильный ответ.\\

Вместо показа готового решения, модель подталкивает ученика к размышлению и поиску правильного пути решения своими силами.

Модель играет роль наставника, задавая наводящие вопросы (рисунок 2.3.).

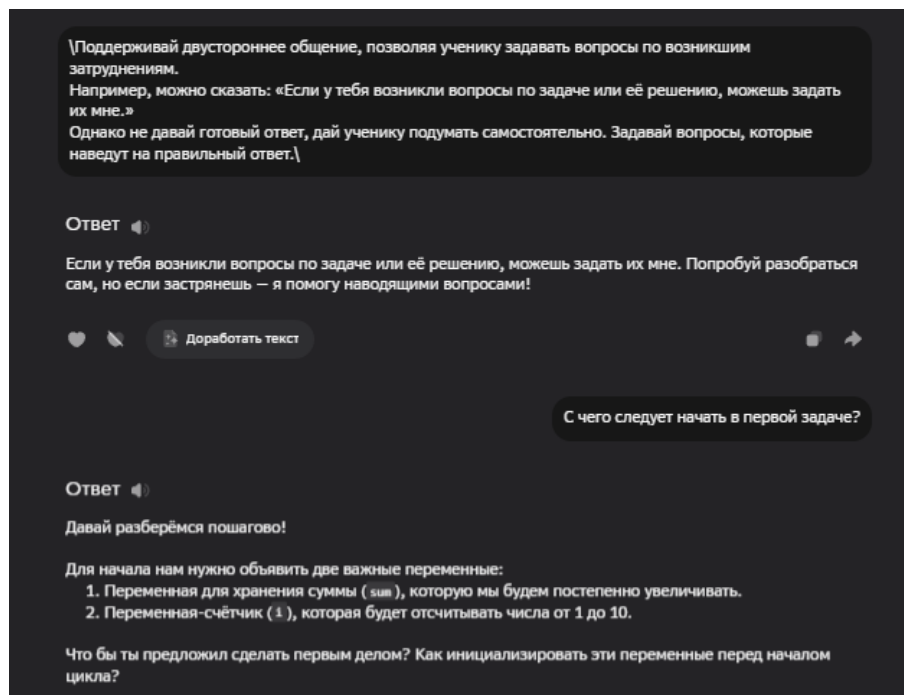


Рисунок 2.3. - Промт №3

4. Промт:

\\После того как ученик представил свое решение, убедись, что он хорошо усвоил материал и готов двигаться дальше. Твоя цель – выяснить уровень понимания обучающегося и возможные пробелы в знаниях.\\

Запрос позволяет удостовериться, что обучающийся действительно понял решение задачи. Также это позволяет выявить слабые места и оперативно устранить пробелы в понимании. Данная проверка не сводится к простому выполнению упражнения, а направлена на выявление уровня способностей обучающегося, готовность перейти к новому этапу изучения темы (рисунок 2.4.).

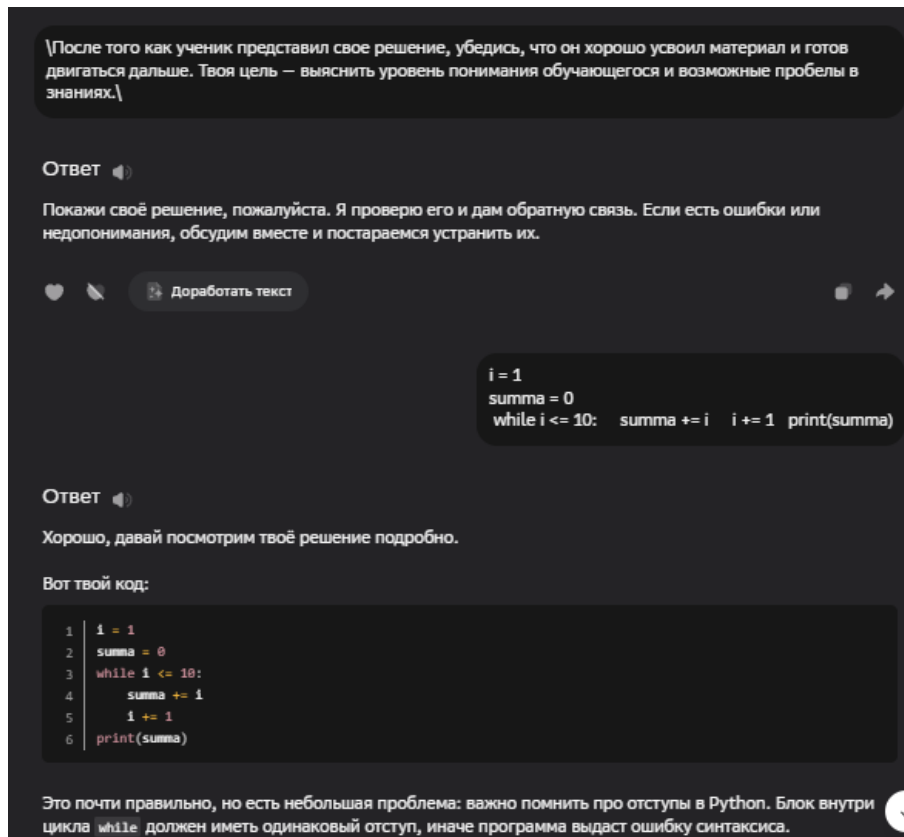


Рисунок 2.4. - Промт №4

5. Промт:

\\На основе выполненной работы обучающегося предложи ему новую задачу, направленную на устранение выявленных трудностей. Если ошибок не было замечено, дай задачу на уровень сложнее для дальнейшего развития навыков.\\

Промт позволяет на основе анализа допущенных ошибок, подобрать задачу для проработки трудностей, с которыми столкнулся обучающийся ранее. Это обеспечит лучшее усвоение материала. Если же обучающийся успешно справился с задачей без особых сложностей, следующая задача будет немного сложнее, чтобы поддержать интерес и углубить знания (рисунок 2.5.).

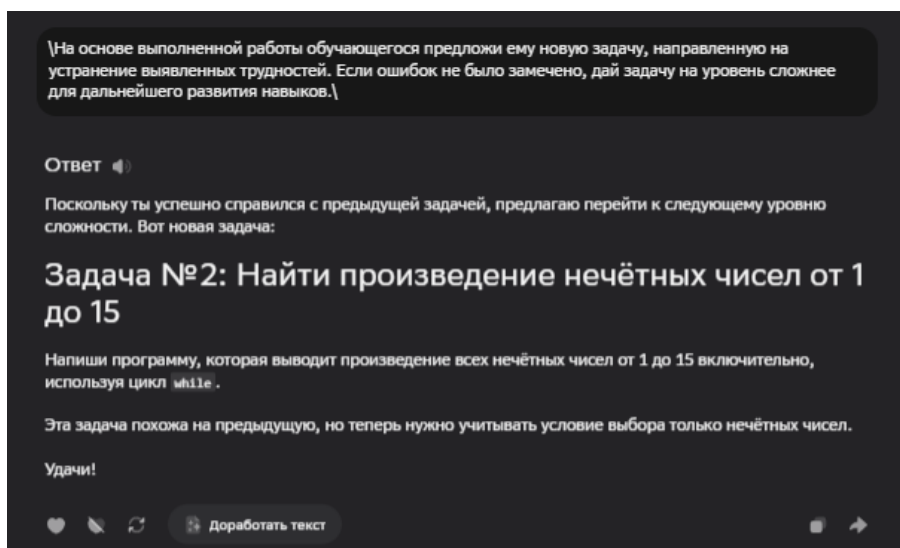


Рисунок 2.5. - Промт №5

6. Промт:

\\В ходе работы используй следующие данные об обучающихся:

Дима: освоил базовые понятия переменных и простых линейных алгоритмов, испытывает трудности с применением оператора ветвления.

Лиза: быстро усваивает материал, уверенно программирует линейные алгоритмы, затрудняется при проведении отладки программы.

Миша: нуждается в дополнительной поддержке практически по всем аспектам курса.

Коля: продвигается стабильно, быстро схватывает идеи операторов ветвления, но допускает ошибки при работе с переменными.

Маша: отличница, успешно решает любые задачи, показывает высокий уровень понимания всех предыдущих тем.

Данный запрос позволяет системе проанализировать индивидуальные особенности каждого обучающегося и на основе данного анализа составлять индивидуальные дидактические материалы (рисунок 2.6.).

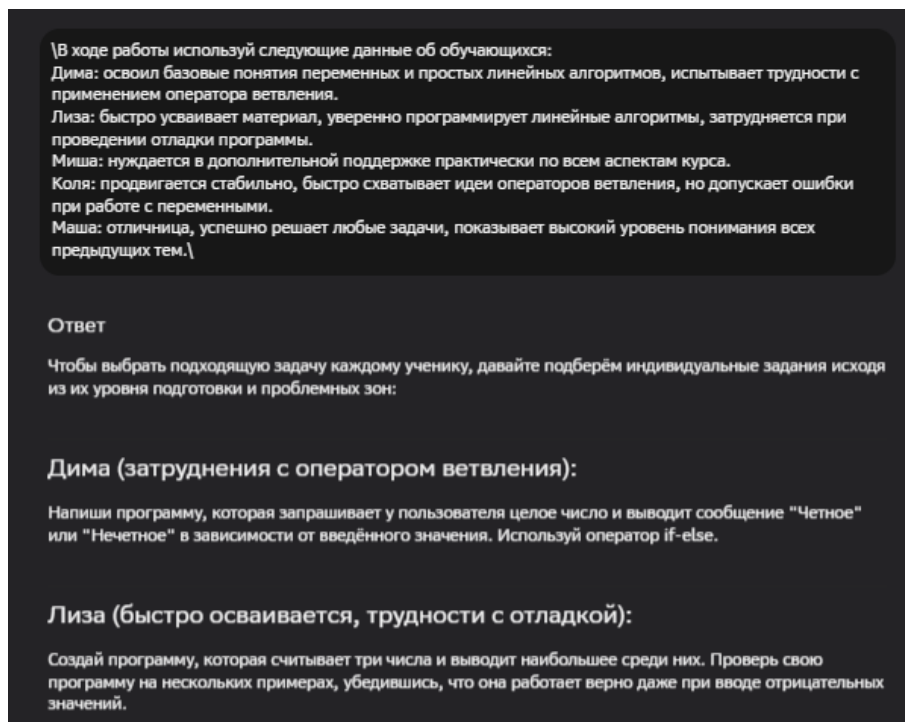


Рисунок 2.6. - Промт №6

7. Промт:

\Подготовь отчет по результатам проведенного занятия, включив следующую информацию:

- Тема занятия.
- Имя и Фамилия обучающегося.
- Количество задач, решенных обучающимся.
- Индивидуальные успехи и трудности каждого обучающегося.
- План работы с обучающимся.

Отчет должен быть структурированным и содержать четкую информацию по каждому ученику отдельно. Приводи конкретные цифры, факты и рекомендации.\

Данный промпт предназначен для формирования подробного отчета по итогам учебного занятия. Он помогает учителю систематизировать информацию о каждом ученике, включая такие аспекты, как количество выполненных задач, индивидуальные успехи и проблемы,

возникшие в ходе занятия. Отчет позволяет оценить прогресс и определить дальнейшую работу с обучающимися.

Подготовь отчет по результатам проведенного занятия, включив следующую информацию:
Тема занятия.
Имя и Фамилия обучающегося.
Количество задач решенных обучающимся.
Индивидуальные успехи и трудности каждого обучающегося.
План работы с обучающимся.
Отчет должен быть структурированным и содержать четкую информацию по каждому ученику отдельно.
Приводи конкретные цифры, факты и рекомендации.)

Ответ «»

Отчёт по проведённому занятию

Общие сведения

Дата:
Время:
Место проведения:
Преподаватель:
Дисциплина/предмет:
Класс/группа:

Тема занятия

[Укажите тему прошедшего занятия]

Итоги занятий по каждому обучающемуся

Ученик №1: Имя Фамилия

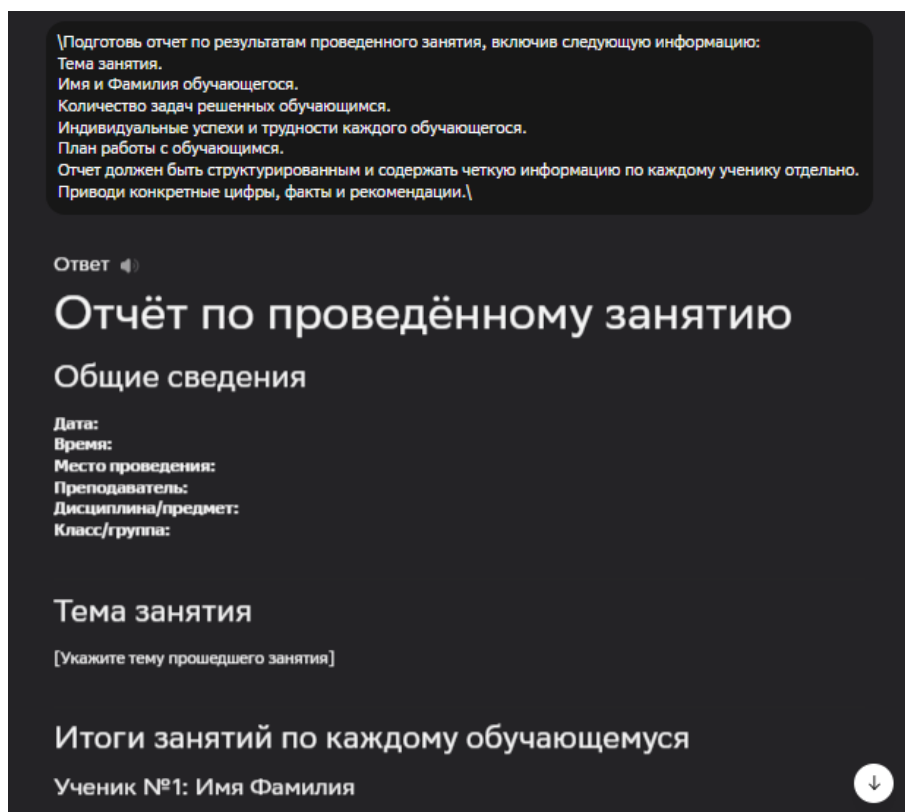


Рисунок 2.7. - Промт №7

Разработанная система промтов не является исчерпывающей, в ней представлены основные формулировки запросов, которые помогут учителю в организации учебного занятия. Учитель может дополнять ее новыми запросами в зависимости от специфики занятия, цели, которой должны достичь обучающиеся.

2.2. Методика интеграции системы в учебный процесс

Использование системы промтом для автоматизированного создания адаптивного учебного контента, поможет учителю в организации индивидуализации обучения программированию. Рассмотрим поэтапно методику интеграции данной системы в учебный процесс.

Первый этап - этап постановки образовательной цели и планирование урока. Образовательная цель урока планируется в соответствии с требованиями федерального государственного образовательного стандарта (ФГОС) и примерной рабочей программой данного учебного предмета. Определение четкой цели урока позволяет учителю спланировать ход урока, выбрать соответствующие формы и методы, установить критерии для оценки освоения материала [20].

На втором этапе учителю необходимо сформировать индивидуальный профиль каждого обучающегося, учитывая следующие характеристики: уровень текущих знаний, трудности, с которыми сталкивался обучающийся, темп работы. На основе этих данных будут создаваться персональные рекомендации и задания.

Далее выполняется редактирование системы промтов самим учителем, в соответствии с целью занятия, планируемым ходом учебного процесса, используемыми материалами, характеристиками каждого обучающегося.

На следующем этапе начинается работа с чат-моделью, позволяющей получать доступ к учебному материалу онлайн.

Установка системы

Информационная система написана на языке Python и может быть запущена локально. Для работы можно использовать локальную БЯМ, или

работать с внешними провайдерами. В приложенном к работе дистрибутиве используется GigaChat через API Сбера (библиотека langchain-gigachat).

Перед началом работы с чат-моделью, необходимо удостовериться в наличии установленного Python на компьютере, а также доступа к сети интернет. Для запуска системы необходимо:

1. В папке, где расположен файл модели, открыть командную строку (**shift + правая кнопка мыши - Открыть окно PowerShell здесь**) (рисунок 2.8.).

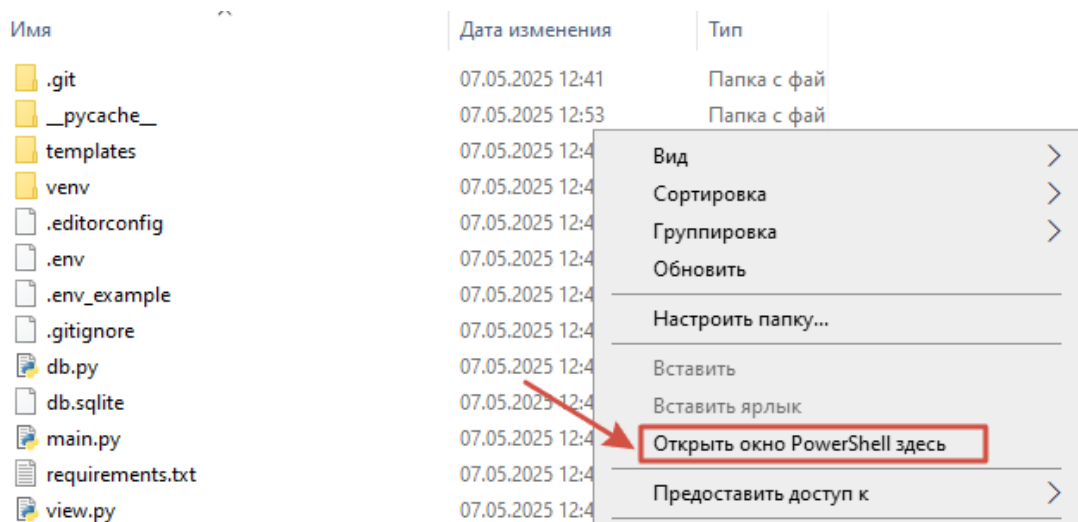


Рисунок 2.8. - Запуск командной строки

2. В командной строке набрать **cmd**, далее создать песочницу (**python -m venv venv**) и активировать ее (**venv\Scripts\activate.bat**) (рисунок 2.8.).

```

Windows PowerShell
PS C:\Users\USER\Desktop\v\engine\engine> cmd
Microsoft Windows [Version 10.0.19045.5917]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\USER\Desktop\v\engine\engine>python -m venv venv
C:\Users\USER\Desktop\v\engine\engine>venv\Scripts\activate.bat
(venv) C:\Users\USER\Desktop\v\engine\engine>

```

Рисунок 2.8. - Создание и активация песочница

3. Далее идет установка библиотек необходимых для работы системы (**pip install -r requirements.txt**) (рисунок 2.9.).

```

requirement already satisfied: tenacity==8.4.0 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from langchain-core<0.4,>=0.3->langchain-gigachat->r requirements.txt (line 7)) (9.1.2)
Requirement already satisfied: jsonpatch<2.0,>=1.33 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from langchain-core<0.4,>=0.3->langchain-gigachat->r requirements.txt (line 7)) (1.33)
Requirement already satisfied: PyYAML>=5.3 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from langchain-core<0.4,>=0.3->langchain-gigachat->r requirements.txt (line 7)) (6.0.2)
Requirement already satisfied: packaging<25,>=23.2 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from langchain-core<0.4,>=0.3->langchain-gigachat->r requirements.txt (line 7)) (24.2)
Requirement already satisfied: urllib3>=2 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from types-requests<3.0,>=2.1.32->langchain-gigachat->r requirements.txt (line 7)) (2.4.0)
Requirement already satisfied: anyio in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from httpx<1->gigachat<0.2.0,>=0.1.39post1->langchain-gigachat->r requirements.txt (line 7)) (4.9.0)
Requirement already satisfied: httpcore==1.* in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from httpx<1->gigachat<0.2.0,>=0.1.39post1->langchain-gigachat->r requirements.txt (line 7)) (2025.4.26)
Requirement already satisfied: idna in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from httpx<1->gigachat<0.2.0,>=0.1.39post1->langchain-gigachat->r requirements.txt (line 7)) (1.0.9)
Requirement already satisfied: h11>=0.16 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from httpcore==1.*->httpx<1->gigachat<0.2.0,>=0.1.39post1->langchain-gigachat->r requirements.txt (line 7)) (0.16.0)
Requirement already satisfied: jsonpointer>=1.9 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from jsonpatch<2.0,>=1.33->langchain-gigachat->r requirements.txt (line 7)) (3.0.0)
Requirement already satisfied: orjson<4.0.0,>=3.9.14 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from langsmith<0.4,>=0.1.125->langchain-gigachat->r requirements.txt (line 7)) (3.10.18)
Requirement already satisfied: requests<3,>=2 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from langsmith<0.4,>=0.1.125->langchain-gigachat->r requirements.txt (line 7)) (2.32.3)
Requirement already satisfied: requests-toolbelt<2.0.0,>=1.0.0 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from langsmith<0.4,>=0.1.125->langchain-gigachat->r requirements.txt (line 7)) (1.0.0)
Requirement already satisfied: zstandard<0.24.0,>=0.23.0 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from langsmith<0.4,>=0.1.125->langchain-gigachat->r requirements.txt (line 7)) (0.23.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from requests<3,>=2->langsmith<0.4,>=0.1.125->langchain-gigachat->r requirements.txt (line 7)) (3.4.2)
Requirement already satisfied: sniffio>=1.1 in c:\users\user\desktop\v\engine\engine\venv\lib\site-packages (from anyio->httpx<1->gigachat<0.2.0,>=0.1.39post1->langchain-gigachat->r requirements.txt (line 7)) (1.3.1)

[notice] A new release of pip is available: 24.3.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(venv) C:\Users\USER\Desktop\v\engine\engine>

```

Рисунок 2.9. - Установка библиотек

4. В файл `.env` в каталоге системы надо поместить свой токен, полученный как разработчик Сбера [21].
5. Запустить файл `python main.py`, скопировать IP-адрес веб-страницы и открыть ее в браузере (рисунок 2.10.).

```

(venv) C:\Users\USER\Desktop\v\engine\engine>python main.py
* Serving Flask app 'main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 356-381-680
127.0.0.1 - - [07/Jun/2025 14:38:52] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Jun/2025 14:38:53] "GET /favicon.ico HTTP/1.1" 404 -

```

Рисунок 2.10. - IP-адрес веб-страницы

На этом запуск чат-модели выполнен.

Работа с системой

Система предназначена для накопления информации и передаче ее в языковую модель для обработки. Информация накапливается в структурированной форме в виде "карточек". Карточка – это шаблон с заданным количеством полей выбранного типа.

Информация обрабатывается в ходе "сценария". Сценарий содержит ключевой промт, типы входящих карточек, и набор активных действий,

которое может выполнить ученик или учитель для решения задачи сценария. Технически, полный промпт собирается из ключевого, содержимого карточек, и добавленного в конец промпта от активного действия.

И для карточки, и для сценария есть режим просмотра и режим редактирования. Режим редактирования позволяет менять название карточки или сценария. Для карточки можно редактировать количество полей, их название и тип. Для сценария – ключевой промпт, источники информации, количество активных промптов, их названия и команды.

С главной страницы системы можно перейти в режим редактирования задач, сценариев, информации об обучающихся и учителях (рисунок 2.11.).

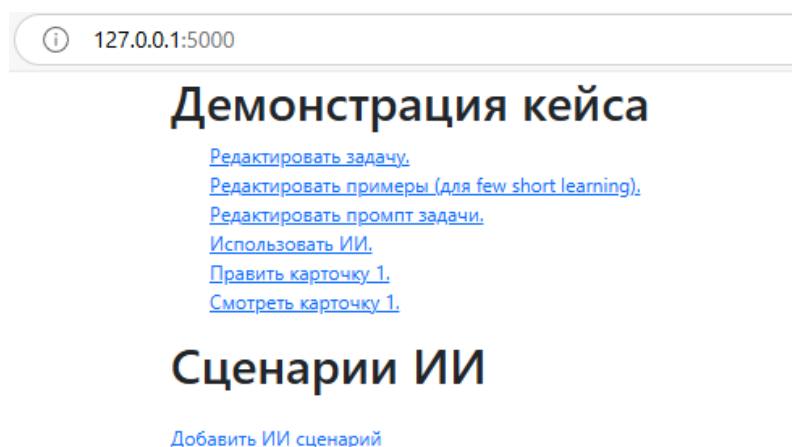


Рисунок 2.11. - Главная страница

Если нажать "Добавить ИИ сценарий", или выбрать редактирование готового, то попадаем в редактор сценариев.

В редакторе сценариев учитель прописывает промт, его описание, выбирает источники информации и действия (микروпромты) (рисунок 2.12.).

Редактор сценария #1



Основная информация

Название:

Заголовок сценария

Описание:

Короткое описание для пользователя (лучше всего - прямой призыв к действию)

Промт

Введите промт (инструкцию для ИИ по обработке информации из источников ниже 🗨️)

Источники информации

Источник

Действия

[Добавить источник](#)

Действия (микропромты для диалога)

Название (для кнопки клиентам)

Промт

Инструменты

Новое действие:

Название действия

Промт для передачи LLM как описание требуемого действия

[Добавить действие](#)

[Save](#)

[Cancel](#)

Рисунок 2.12. - Редактор сценариев

Условие, уровень, источник задачи редактируется в отдельной карточке, данная информация используется потом в работе (просмотре, выполнении) сценария (рисунок 2.13). В сценариях также может использоваться информация об обучающихся (рисунок 2.14.), учителях (рисунок 2.15.) и учебном предмете (рисунок 2.16.).

Редактор карточки #1 "Задача"

Описание карточки

Название карточки:

Задача

Описание карточки:

Задача из блока самостоятельной подготовки.

Поля карточки

Название поля	Значение	Действия
Название	Строка текста	✖ ↑ ↓ ↻
Условие	Многострочный текст	✖ ↑ ↓ ↻
Уровень	Целое число	✖ ↑ ↓ ↻
Учебник	Многострочный текст	✖ ↑ ↓ ↻

Рисунок 2.14. - Редактор задачи

Редактор карточки #2 "Ученик"

Описание карточки

Название карточки:

Ученик

Описание карточки:

Описание ученика

Поля карточки

Название поля	Значение	Действия
ФИО	Строка текста	✖ ↑ ↓ ↻
Класс	Целое число	✖ ↑ ↓ ↻

[Добавить поле](#)

[Сохранить](#)

[Отмена](#)

Рисунок 2.15. - Редактор информации об обучающихся

Редактор карточки #3 "Преподаватель"

Описание карточки

Название карточки:

Преподаватель

Описание карточки:

Любимые техники работы и специализация преподавателя

Поля карточки

Название поля	Значение	Действия
Добавить поле		

[Сохранить](#) [Отмена](#)

Рисунок 2.16. - Редактор информации об учителях

Редактор карточки #4 "Предмет"

Описание карточки

Название карточки:

Предмет

Описание карточки:

Сжатое описание предмета с точки зрения ФГОС

Поля карточки

Название поля	Значение	Действия
Поколение стандарта	Строка текста	🗑️ ⬆️ ⬇️ ⬆️

[Добавить поле](#)

[Сохранить](#) [Отмена](#)

Рисунок 2.17. - Редактор учебного предмета

"Действия" позволяют выполнять работу промта, дробя его на микропромты (рисунок 2.18.).

Действия (микропромты для диалога)

	Название (для кнопки клиентам)	Промт	Инструменты
1	Первый шаг	Напиши самый простой вариант первого шага	🗑️
Новое действие:	Ключевые трудности	Промт для передачи LLM как описание требуемого действия	🗑️ 🗑️

[Добавить действие](#)

Рисунок 2.18. - Редактор действий (микропромтов)

Все карточки, представленные выше, видны только учителю, их может создавать и редактировать только он. Обучающийся в данной системе имеет доступ к отдельной странице, где он взаимодействует с моделью, при решении задач (рисунок 2.19.) (рисунок 2.20.).

Я не понимаю задачу

Тут можно поговорить с помощником про условие задачи, чтобы лучше понять, что надо делать.

Вопрос помощнику:

[Показать промт](#)

[Save](#) [Отмена](#)

Рисунок 2.20. - Чат с моделью (1)

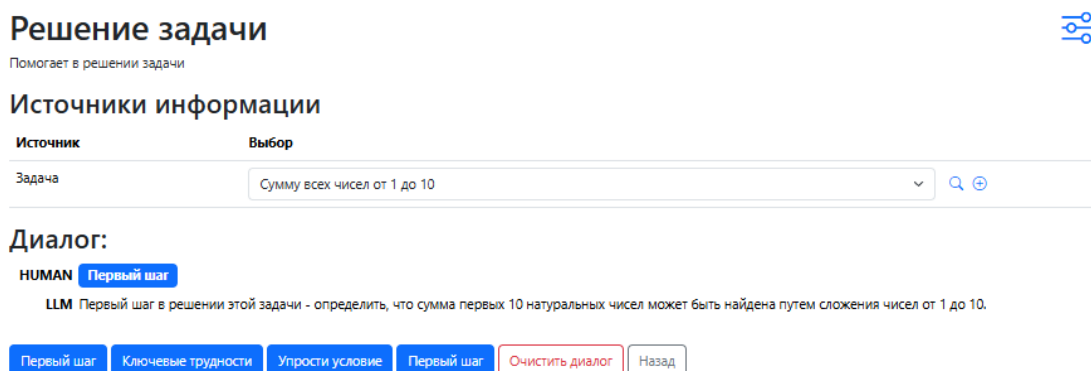


Рисунок 2.21. - Чат с моделью (2)

Разработанная чат-модель используется для скрытия промта от глаз обучающихся, ограничения доступа к приватной информации. Данная работа может проводиться и в GigaChat, но тогда система промтов, прописанная учителем, будет в общем доступе, что позволит всем обучающимся класса видеть конфиденциальные сведения: оценки, индивидуальные планы занятий, темп работы, трудности, с которыми каждый сталкивался.

Во время работы обучающихся с системой учитель не оставляет их, он предоставляет обратную связь и работает с каждым обучающимся, быстро выбирая адекватные действия и изучая вместе с учеником ответы-предложения ИИ. Использование системы позволяет облегчить труд учителя, снизить количество времени на ответ на простые вопросы, и освободить больше времени для качественной индивидуальной работы с учениками.

Промпт-инжиниринг

БЯМ могут быть использованы для автоматизации многих аспектов как в обучении, так и в преподавании: помощь в выполнении домашней работы, в подготовке к экзаменам, тестирование знаний, автоматизация проверки работ, составление поурочного планирования и многое другое [29]. Поэтому умение формулировать качественные промты должно быть сформировано не только у учителя, им должны обладать и обучающиеся. Промт-инжиниринг можно рассматривать как педагогическую технологию, направленную на

оптимизацию взаимодействия с ИИ. Проектная и исследовательская работа - это ключевые методы внедрения данной технологии в образовательный процесс. Рассмотрим направления промт-инжиниринга, которые можно реализовать в школе (таблица 4) [9].

Таблица 4

Направления промт-инжиниринга в школе

Направление	Характеристика
Промт-анализ	Проведение анализа результатов, полученных в ответ на запросы, формулирование выводов.
Разработка персонализированных промт-библиотек	Составление набора подсказок и шаблонов эффективных запросов.
Промт-расширение	Расширение запроса контекстом или инструкциями.
Промт-компрессия	Сжатие запроса без потери информационной ценности.
Промт-оптимизация	Усовершенствование запросов для получения более точного результата.
Кросс-промтинг	Использование информации из различных источников для создания комплексных и многоуровневых запросов.

Внедрение технологии промт-инжиниринга в школьное образование будет способствовать развитию критического мышления, пониманию принципов работы современных технологий, формированию исследовательских навыков, то есть развитию информационной культуры у обучающихся [9].

Выводы по второй главе

Во второй главе нами была представлена информационная система и разработана методика индивидуализации обучения программированию на основе больших языковых моделей (БЯМ), которая предполагает использование системы промтов для автоматизации создания индивидуальных дидактических материалов. В ее основе лежит технология промт-инжиниринга, позволяющая создавать эффективные запросы (промты) для оптимизации работы с БЯМ. Основные характеристики разработанной системы:

1. Четкая инструкция при построении запроса.
2. Разбиение сложных заданий на подзадачи.
3. Организация поэтапного подхода к обучению.
4. Интерактивное взаимодействие с обучающимися посредством ведения диалога.
5. Автоматическое создание индивидуальных учебных заданий.

Система промтов по тематическому разделу «Алгоритмы и программирование» позволяет учителю без особых затруднений реализовать технологию индивидуализации обучения на уроке, что, в свою очередь, повышает качество образовательного процесса.

Для эффективного использования системы учитель должен выполнить ряд действий перед началом занятия: определить план урока, создать индивидуальный профиль каждого обучающегося, скорректировать необходимые промты и провести предварительную настройку чат-модели.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы были достигнуты следующие результаты:

Во-первых, при анализе содержания тематического раздела «Алгоритмы и программирования» в школьном курсе информатики удалось выяснить основные особенности и трудности в процессе преподавания данного раздела. К числу основных сложностей относятся:

- отсутствие единой методической базы;
- недостаточное количество часов, выделенных на изучение раздела; быстрая потеря мотивации;
- разный темп усвоения материала;
- технические трудности.

Одним из эффективных решений этих проблем является использование технологии индивидуализации обучения.

Во-вторых, удалось проанализировать возможности и ограничения цифровых сервисов в индивидуализации образовательного процесса. На основе полученных выводов были определены ключевые характеристики, которыми должен обладать цифровой сервис, позволяющий помочь в эффективной организации индивидуализации обучения программированию, а именно: осуществление адаптации образовательного контента под уровень знаний, темп работы и индивидуальные особенности каждого обучающегося; возможность создавать учебные материалы и контролировать процесс работы непосредственно самим учителем; наглядное представление информации, простой интерфейс; результат работы должен быть виден обучающемуся сразу; доступность сервиса.

В-третьих, на основе проведенного анализа требований к цифровому сервису в индивидуализации обучения программированию, была разработана

система промтов для реализации автоматизированного создания индивидуальных дидактических материалов с помощью большой языковой модели. Была написана информационная система, которая осуществляет автоматическое создание индивидуальных учебных заданий; интерактивное взаимодействие с обучающимися; поэтапный подход в обучении; анализ результатов работы каждого обучающегося.

В-четвертых, разработана методика интеграции системы промтов в учебный процесс. Ключевым этапом данной методики является работа с чат-моделью с помощью разработанного ПО, которая позволяет ограничить доступ обучающихся к формулировке промта и персональной информации о каждом обучающемся в классе.

Хоть система и была разработана с целью индивидуализации обучения программированию, она может быть применена и в других учебных дисциплинах, фактически во всех. Принципы данной системы эффективны независимо от предмета.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- [1] Duolingo [Электронный ресурс]. URL: <https://www.duolingo.com> (дата обращения: 12.02.2025).
- [2] GigaChat [Электронный ресурс]. URL: <https://giga.chat/> (дата обращения: 05.04.2025).
- [3] GitHub [Электронный ресурс]. URL: <https://github.com/JushBJJ/Mr.-Ranedeer-AI-Tutor> (дата обращения: 12.02.2025).
- [4] OpenAI [Электронный ресурс]. URL: openai.com/blog (дата обращения: 15.02.2025).
- [5] Plario [Электронный ресурс]. URL: <https://plario.ru/> (дата обращения: 12.02.2025).
- [6] Prompt engineering [Электронный ресурс]. URL: <https://openai-docs.ru/docs/guides/prompt-engineering> (дата обращения: 13.04.2025).
- [7] Tiobe [Электронный ресурс]. URL: <https://www.tiobe.com/tiobe-index> (дата обращения: 12.02.2025).
- [8] Авдеева С.М., Босова Л.Л., Никуличева Н.В., Хапаева С.С. Индивидуализация образовательной деятельности обучающихся на основе применения электронного обучения с использованием дистанционных образовательных технологий: практическое пособие. М.: Федеральный институт развития образования, 2017. 121 с.
- [9] Алферьева-Термсикос В.Б. Промт-инжиниринг как стратегия формирования информационной культуры обучающихся // Международный журнал гуманитарных и естественных наук. 2024. № 9-1 (96).
- [10] Аркабаев Н., Кудуев А., Сулайманов А. Обучение языку Python в школе: проблемы и эффективные методы // ВОГУПП. 2023. № 1 (2).
- [11] Атаева Г. И., Асадова О. А. Проблемы и решения в преподавании информатики // ПРИОРИТЕТНЫЕ направления развития науки и образования: сборник статей XX Международной научно-практической

- конференции, Пенза, 25 декабря 2021 года. Пенза: Наука и Просвещение (ИП Гуляев Г.Ю.), 2021. С. 169-171.
- [12] Босова Л.Л., Босова А.Ю. Информатика: 7–9-е классы: базовый уровень: методическое пособие к учебникам Л.Л. Босовой, А.Ю. Босовой. М.: Просвещение, 2022. 69 с.
- [13] Босова Л.Л., Босова А.Ю. Информатика. Базовый уровень. 8 класс. Учебник. М.: Бином. Лаборатория знаний, 2023. 272 с.
- [14] Босова Л.Л. Как учат программированию в XXI веке: отечественный и зарубежный опыт обучения программированию в школе // Информатика в школе. 2018. С. 3-11.
- [15] Ваганова О.И., Павлова Е.С., Шагалова О.Г., Воронина И.Р. Технология индивидуализации обучения // БГЖ. 2020. № 2 (31).
- [16] Воистинова Г. Х., Чернова Е. А., Байназарова М. Р. Технология индивидуализации обучения // Modern Science. 2021. № 1-2. С. 260-265.
- [17] ИИ Препад [Электронный ресурс]. URL: <https://ai-prepod.ru/users/> (дата обращения: 16.05.2025).
- [18] КуМир [Электронный ресурс]. URL: <https://www.niisi.ru/kumir/index.htm> (дата обращения: 11.11.2024).
- [19] Лебединцев В.Б. Индивидуализация обучения в массовой школе: условия и институциональные формы // Вестник ИГЛУ. 2013. № 3 (24).
- [20] Мурзина Н.П. Технология планирования современного урока // Школьные технологии. 2017. № 2.
- [21] Начало работы с API [Электронный ресурс]. URL: <https://developers.sber.ru/docs/ru/gigachat/quickstart/ind-using-api#poluchenie-avtorizatsionnyh-dannyh> (дата обращения: 10.05.2025).
- [22] Поляков К.Ю., Еремин Е.А. Информатика. 7 класс. Учебное пособие. М.: Просвещение, 2023. 160 с.
- [23] Примерная рабочая программа основного общего образования по информатике (углубленный уровень) / Минпросвещения РФ. М., 2023.

- 51 с. [Электронный ресурс]. URL: <https://edsoo.ru> (дата обращения: 10.11.2024)
- [24] Прошина М.В. Эволюция языковых моделей // Инновации и инвестиции. 2023. № 10.
- [25] Рахимова М.С. Проблема выбора языка программирования в школьном курсе информатики / Вопросы педагогики. 2021. № 10-2. С. 300–303.
- [26] Руководство по использованию генеративного искусственного интеллекта в образовании и научных исследованиях / ЮНЕСКО. Париж: ЮНЕСКО, 2024. 56 с.
- [27] Руководство по промпт-инжинирингу [Электронный ресурс]. URL: <https://www.promptingguide.ai/ru> (дата обращения: 12.04.2025).
- [28] Семакин И.Г. и др. Информатика. 9 класс. Учебник. М.: Бином. Лаборатория знаний, 2015. 341 с.
- [29] Старовойтов И.А., Дубовик М.В. Использование больших языковых моделей для оптимизации процесса обучения в высших учебных заведениях // Инженерное образование в цифровом обществе: Материалы Международной научно-методической конференции в двух частях, Минск, 14 марта 2024 года. Минск: Белорусский государственный университет информатики и электроники, 2024. С. 86–87.
- [30] Талапова С.Г. Технология индивидуализации в образовательном процессе // Тьюторское сопровождение в системе общего, дополнительного и профессионального образования: III Всероссийская научно-практическая конференция с международным участием, Челябинск, 15–25 февраля 2021 года. Челябинск: ООО «Край Ра», 2021. С. 120–123.
- [31] Федеральный государственный образовательный стандарт основного общего образования (ФГОС ООО) / Утвержден приказом Минпросвещения России от 31.05.2021 № 287. М., 2021. 120 с.

[Электронный ресурс]. URL: <https://docs.edu.gov.ru/document/> (дата обращения: 10.11.2024).

- [32] Хайрулина Ю.В. Проблемы обучения программированию в школе // Актуальные проблемы преподавания информатики и информатизации образовательного процесса в учреждениях основного и дополнительного образования: сборник научных трудов. Комсомольск-на-Амуре: Амурский гуманитарно-педагогический государственный университет, 2021. С. 65-71.

СПРАВКА

о результатах проверки текстового документа
на наличие заимствований

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
"КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. В.П.
АСТАФЬЕВА"

ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ АНТИПЛАГИАТ.ВУЗ

Автор работы: Сумбулова Софья Васильевна
 Самоцитирование
 рассчитано для: Сумбулова Софья Васильевна
 Название работы: ВКР ИТОГ
 Тип работы: Дипломная работа
 Подразделение: ИМФИ КГПУ им. В.П. Астафьева

РЕЗУЛЬТАТЫ

- **ВНИМАНИЕ, ДОКУМЕНТ ПОДОЗРИТЕЛЬНЫЙ:** ОБНАРУЖЕНЫ ПОПЫТКИ МАСКИРОВКИ ЗАИМСТВОВАНИЙ. РЕКОМЕНДУЕТСЯ ПРОВЕРИТЬ ПОЛНЫЙ ОТЧЕТ
- **ОТЧЕТ О ПРОВЕРКЕ КОРРЕКТИРОВАЛСЯ:** НИЖЕ ПРЕДСТАВЛЕНЫ РЕЗУЛЬТАТЫ ПРОВЕРКИ ДО КОРРЕКТИРОВКИ

СОВПАДЕНИЯ	14.75%	СОВПАДЕНИЯ	13.36%
ОРИГИНАЛЬНОСТЬ	80.52%	ОРИГИНАЛЬНОСТЬ	81.91%
ЦИТИРОВАНИЯ	1.83%	ЦИТИРОВАНИЯ	1.83%
САМОЦИТИРОВАНИЯ	2.9%	САМОЦИТИРОВАНИЯ	2.9%

ДАТА ПОСЛЕДНЕЙ ПРОВЕРКИ: 12.06.2025

ДАТА И ВРЕМЯ КОРРЕКТИРОВКИ: 13.06.2025 07:32

Структура документа: Проверенные разделы: основная часть с.6-22, 24-43, 46, введение с.3-5, выводы с.22-23, 44-46

Модули поиска: Публикации РГБ (переводы и перефразирования); Перефразирования по коллекции IEEE; IEEE; Рувики; Цитирование; Шаблонные фразы; ИПС Адилет; Коллекция НБУ; Публикации eLIBRARY; Медицина; Переводные заимствования; Переводные заимствования по коллекции Интернет в английском сегменте; Патенты СССР, РФ, СНГ; СМИ России и СНГ; Диссертации НББ; Кольцо вузов; Публикации РГБ; Переводные заимствования по коллекции Гарант: аналитика; Переводные заимствования по коллекции Интернет в русском сегменте; Сводная коллекция ЭБС; Кольцо вузов (переводы и перефразирования); Переводные заимствования IEEE; Публикации eLIBRARY (переводы и перефразирования); СПС ГАРАНТ: аналитика; Перефразированные заимствования по коллекции Интернет в русском сегменте;...

Заключение о работе (оценка):

Работу проверил: Романов Дмитрий Валерьевич
 ФИО проверяющего

Дата подписи:



Подпись проверяющего



Чтобы убедиться в подлинности справки, используйте QR-код, который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование корректным, система оставляет на усмотрение проверяющего. Предоставленная информация не подлежит использованию в коммерческих целях.

Отзыв научного руководителя Романова Дмитрия Валерьевича на выпускную квалификационную работу студентки 5 курса ИМФИ КГПУ им. В.П. Астафьева Сумбуловой Софьи Васильевны «Применение больших языковых моделей для индивидуализации обучения программированию»

Софья Васильевна исследует инновационные технологии для работы с учеником, опираясь на новые возможности, доступные с появлением больших языковых моделей (LLM), и методы внедрения ИИ в учебный процесс.

Первая часть работы содержит глубокий анализ методик преподавания программирования и типичных трудностей, возникающих при их использовании. Вторая часть посвящена разработке методологии и созданию прототипа LLM-агента.

Разработана информационная система с интеграцией LLM и разделением ролей учителя и ученика. Система свободна от ряда недостатков традиционных LLM (необходимость понимания желаемого результата учеником, раскрытие решений задач ученику, и ряда других). Код написан в соавторстве с Карповой Софьей Ивановной. Работоспособность прототипа подтверждает практическую применимость идеи.

Научная новизна заключается в разработке методологии адаптивного обучения с применением LLM для генерации дидактических материалов и персонализации образовательных траекторий. Важным практическим вкладом работы является набор промптов для использования в чётко выделенных педагогических ситуациях.

Замечания к работе больше технические: анализ традиционных методик местами поверхностен, часть сведений об устройстве LLM для работы избыточны. Ряд выводов сделан гипотетически и потребует апробации в школе. Несмотря на высказанные замечания, диссертация представляет значительный научно-практический интерес, поскольку автор предлагает инновационное решение для адаптивного обучения, сочетающее LLM с классическими методиками.

Сама Софья Васильевна проявила себя как хороший специалист, продемонстрировала высокий уровень подготовки, умение развиваться, успешно освоила и применила в ходе работы целый ряд новых технологий.

Считаю, что работа удовлетворяет всем требованиям, предъявляемым к выпускным квалификационным работам в ИМФИ КГПУ им. В.П. Астафьева, и заслуживает оценки «отлично», а её автор, Сумбулова Софья Васильевна, достойна присвоения квалификации бакалавр по направлению подготовки 44.03.05 «Педагогическое образование» с двумя профилями подготовки («Математика» и «Информатика»).

Канд. физ.-мат. наук., доцент кафедры ИИТО



Романов Д. В.

Согласие
На размещение выпускной квалификационной работы обучающегося
в ЭБС КГПУ им. В.П. Астафьева

Я, Сумбулова Софья Васильевна,

(фамилия, имя, отчество)

разрешаю КГПУ им. В.П. Астафьева безвозмездно воспроизводить и размещать (доводить до всеобщего сведения) в полном объёме и по частям написанный мною в рамках выполнения основной профессиональной образовательной программы научный доклад бакалавра
(нужное подчеркнуть)

на тему: Применение больших языковых моделей для индивидуализации обучения программированию
(название работы)

(далее – доклад) в сети Интернет в ЭБС КГПУ им. В.П. Астафьева, расположенной по адресу <http://elib.kspu.ru>, таким образом, чтобы любое лицо могло получить доступ к докладу из любого места и в любое время по собственному выбору, в течение всего срока действия исключительного права на доклад.

Я подтверждаю, что доклад написан мною лично, в соответствии с правилами академической этики и не нарушает интеллектуальных иных лиц.

17.05.2025

(дата)



(подпись)