

Рекомендации для системы образования по совершенствованию методики преподавания информатики на основе анализа выполнения учащимися заданий КИМ ЕГЭ 2026 и выявленных типичных ошибок и затруднений

Рекомендации для учителей по совершенствованию преподавания учебного предмета группе обучающихся с минимальным уровнем подготовки

Системный подход к преподаванию информатики обеспечивает постепенное накопление и структурирование знаний, умений и навыков, необходимых для освоения школьной программы, что особенно значимо для обучающихся с минимальным уровнем подготовки, у которых пробелы в базовых представлениях препятствуют успешному усвоению материала на более высоких уровнях сложности. В связи с этим при планировании и организации учебного процесса для данной группы обучающихся рекомендуем учитывать следующие особенности, ориентированные на формирование предметных и метапредметных результатов.

1. Систематическая отработка базовых алгоритмических тем на уроках в 8–9 классах. Учащимся с минимальным уровнем подготовки сложно воспринимать алгоритмы в абстрактном виде, поэтому необходимо перенести акцент с понимания логики на формальное выполнение инструкций. На каждом уроке в 8–9 классах целесообразно выделять 5–7 минут на трассировку коротких линейных и разветвляющихся алгоритмов — это возможно сделать в рамках организационного момента или в качестве разминки в начале урока. Обязательным элементом должно стать ведение таблицы трассировки, в которой фиксируются все изменения значений переменных на каждом шаге, что делает алгоритм «видимым» и превращает его в понятную последовательность действий. Использование цветового выделения ячеек таблицы помогает наглядно отслеживать изменения и подходит для любых условий работы (доска, тетрадь, проектор). Начинать рекомендуется с простейших примеров с 1–2 переменными и 3–4 шагами, постепенно усложняя и добавляя ветвления. Такой подход позволяет отработать задания 5, 6, 12 и 23 без дополнительных временных затрат.

2. Регулярные практикумы по работе с электронными таблицами в 9 и 11 классах. Учащиеся с минимальным уровнем подготовки часто теряются в интерфейсе и путают функции, потому что работают с таблицами эпизодически — необходима регулярная краткая практика, встроенная в урочную деятельность. Рекомендуется проводить еженедельные 15-минутные практикумы, которые можно организовать в конце урока, когда основная тема уже пройдена, или выделить для них отдельные уроки в рамках повторения. Использование открытых форматов файлов (*.ods) направлено на формирование умений к работе в разных средах. Каждое занятие рекомендуется посвящать одной функции (СУММ, СРЗНАЧ, МИН, МАКС, СЧЁТЕСЛИ), но отрабатывать её на реальных небольших наборах данных из 5–10 строк, максимально приближенных к экзаменационным форматам. Важно давать чёткий шаблон выполнения, состоящий из нескольких шагов: прочитать условие, определить, какая функция нужна, выделить диапазон, записать формулу, проверить результат на

контрольном примере. Постепенно это позволяет сформировать устойчивый навык для выполнения задания 9.

3. Поэтапное формирование понятий алгебры логики в 8 и 10 классах. Логические выражения с несколькими переменными представляют для слабо подготовленных учащихся значительную сложность, поскольку у них отсутствует устойчивое знание даже таблиц истинности простых операций. Обучение должно начинаться в 8 классе с построения таблиц для одной операции (НЕ, И, ИЛИ) с обязательным использованием цветовой схемы, где каждый тип операции выделяется своим цветом, что особенно удобно при работе у доски или в тетради. Только после уверенного освоения простых операций следует переходить в 10 классе к составным выражениям с 2–3 переменными, причём для слабо подготовленных учащихся целесообразно предоставлять готовые шаблоны таблиц с уже заполненными наборами значений переменных, чтобы они сосредоточились исключительно на вычислении значений выражений.

4. Формирование навыка самопроверки на основе контрольных примеров. Главная проблема учащихся с минимальным уровнем подготовки — они не проверяют себя, и даже если решение неверное, они сдают работу, потому что не знают, как проверить ответ. В связи с этим рекомендуется ввести обязательное правило: при решении любой задачи учащийся должен записать 2–3 проверочных набора входных данных и ожидаемый результат. На начальном этапе можно давать уже готовые проверочные наборы, а затем постепенно переходить к самостоятельному их подбору. Этот навык полезен при решении любых заданий, но особенно критичен для заданий с развёрнутым ответом, где проверка на контрольных примерах помогает избежать грубых ошибок. Сделать проверку на примерах обязательным этапом решения и учитывать её при оценивании (например, добавлять бонусный балл за правильно подобранные проверочные данные) — вполне реалистичное решение в рамках школьной оценочной системы.

5. Систематическая работа с текстовой информацией. У обучающихся отмечается недостаточный уровень владения инструментами обработки текстовой информации, что обусловлено эпизодическим характером соответствующей практики в рамках учебного процесса. Для формирования устойчивых навыков работы с текстовыми данными рекомендуется включать задания на поиск и замену подстрок, а также на подсчёт количества вхождений слова в текст в структуру еженедельных мини-практикумов по аналогии с практикумами по электронным таблицам, что не требует выделения дополнительного учебного времени при условии интеграции данных заданий в уже существующие уроки. В качестве дидактического материала целесообразно использовать аутентичные тексты (фрагменты учебников, новостные сообщения, инструктивные материалы), что способствует повышению познавательной мотивации и демонстрации практической значимости формируемого навыка. Отработка умений может осуществляться различными способами: визуальный анализ текста, применение встроенных функций текстовых редакторов, использование простых программных средств при наличии соответствующих технических условий. На начальном этапе рекомендуется использовать короткие тексты объёмом 1–2

абзаца и задания одного типа, постепенно увеличивая уровень сложности. Данный подход позволяет подготовить обучающихся к выполнению задания 10 контрольных измерительных материалов без дополнительной временной нагрузки на урок.

6.Формирование навыка чтения и интерпретации программного кода без использования компьютера. Для слабоуспевающих учащихся значительную трудность представляет выполнение заданий, связанных с анализом готовых программных фрагментов, поскольку у них не сформирован навык «ручной трассировки» алгоритма. Рекомендуется начиная с 8 класса систематически включать в уроки задания на чтение коротких программ, записанных на изучаемом языке программирования, с обязательным выполнением пошаговой трассировки в тетради без использования компилятора, что позволяет сформировать устойчивое понимание логики работы условных операторов и циклов.

7.Освоение комбинированных методов решения с использованием нескольких инструментов. Рекомендуется задания выполнять двумя способами, например, с помощью электронных таблиц, так и с использованием языков программирования. Такой подход позволяет сформировать у учащихся понимание, что многие задачи имеют альтернативные методы решения, и выбор наиболее эффективного из них зависит от конкретной ситуации и индивидуальных предпочтений. Для слабо подготовленных учащихся это также является дополнительной возможностью проверки полученного результата: если два способа дают одинаковый ответ, его достоверность существенно возрастает.

8.Методика организации коррекционной работы с обучающимися минимального уровня подготовки. Эффективность подготовки данной группы обучающихся достигается при условии адресной коррекционной работы на каждом уроке. Целесообразно выделять некоторое время (например, 5-7 минут) в начале урока для индивидуальных тренировочных упражнений по карточкам, направленных на устранение конкретных дефицитов. Освоение каждого нового типа заданий должно сопровождаться разработкой алгоритмических предписаний и пошаговых инструкций, которые служат опорой на этапе формирования навыка, но постепенно сокращаются по мере его автоматизации. Для развития читательской грамотности требуется систематическая практика смыслового анализа условия: выделение ключевых слов, переформулирование вопроса, построение краткой записи. Формирование регулятивных универсальных учебных действий предполагает организацию самоконтроля и рефлексии через вопросы о способах проверки ответа и выявлении возможных ошибок, а также через фиксацию типичных затруднений и способов их преодоления.

9. Применение цифровых ресурсов. В процессе обучения целесообразно применять цифровые ресурсы (например, Яндекс Учебник, ФГИС «Моя школа») с функцией обратной связи, что особенно значимо для слабо подготовленных учащихся. Применение визуальных средств представления информации позволяет перевести абстрактное содержание в наглядно-образную форму, что способствует его осмысленному усвоению и снижает когнитивную нагрузку на обучающихся. Включение в домашние задания коротких интерактивных упражнений с

автоматической проверкой позволяет многократно тренироваться без риска получить негативную оценку, снижает тревожность и повышает мотивацию.

Рекомендации для учителей по совершенствованию преподавания учебного предмета группе обучающихся с низким уровнем подготовки

Системный подход к преподаванию информатики обеспечивает постепенное накопление и структурирование знаний, умений и навыков, необходимых для освоения школьной программы, что особенно значимо для обучающихся с минимальным уровнем подготовки, у которых пробелы в базовых представлениях препятствуют успешному усвоению материала на более высоких уровнях сложности. В связи с этим при планировании и организации учебного процесса для данной группы обучающихся рекомендуем учитывать следующие особенности, ориентированные на формирование предметных и метапредметных результатов.

1. Ликвидация пробелов в темах «Измерение количества информации» и «Позиционные системы счисления». Типичные ошибки связаны с неверным применением формулы Хартли и нарушением алгоритмов перевода чисел. Рекомендуется отказаться от формального заучивания формулы $N = 2^i$, используя визуализацию (деревья вариантов, таблицы кодирования). При изучении систем счисления систематически отрабатывать перевод чисел в пределах 0–255 с обязательной проверкой обратным переводом. Эффективен приём парной работы: один ученик переводит число, другой выполняет обратное преобразование.

2. Формирование устойчивых навыков работы с электронными таблицами. Учащиеся воспринимают таблицы как средство хранения, но не как вычислительный инструмент. Рекомендуется изучать функции через решение практических задач (например, статистическая обработка, подсчёт успеваемости и т.д.). Освоение функций осуществлять поэтапно: от готовых формул с пропусками к самостоятельной записи СУММ, СРЗНАЧ, СЧЁТЕСЛИ, МИН, МАКС. Особое внимание — синтаксису критерия в СЧЁТЕСЛИ. Обучать самопроверке через сравнение результата функции с ручным подсчётом.

3. Развитие умений анализа и формального исполнения алгоритмов. Основной дефицит — неумение читать готовый код и проследивать изменение переменных. Рекомендуется акцент делать на «чтении» алгоритмов: восстановление пропущенных фрагментов, определение результата при заданных данных, поиск ошибок в готовом коде. Широко использовать таблицы трассировки: на начальном этапе заполнять совместно с учителем, затем самостоятельно. Для отработки рекуррентных соотношений рекомендуется требовать записи не только конечного результата, но и всех промежуточных значений последовательности, что позволяет проследить логику вычислений. Эффективным приёмом является использование заданий с изменёнными условиями (например, изменение знака сравнения или начального значения переменной), что заставляет учащихся перестраивать ход рассуждений и не действовать по шаблону. Также полезно включать задания, в которых необходимо не вычислить результат, а определить, при каких входных данных алгоритм даст

указанный результат (обратная задача), что развивает понимание логики работы алгоритма.

4.Преодоление трудностей при изучении адресации в компьютерных сетях. Типичные ошибки — неверный перевод чисел и непонимание поразрядной конъюнкции. Рекомендуется визуализировать IP-адрес и маску как строки битов с цветовым выделением разрядов. Полезны задания на обратный ход: восстановление маски по нескольким адресам одной сети. Тема изучается только после освоения двоичной арифметики.

5.Освоение основ математической логики. Рекомендуется начинать с простых высказываний, переходя к составным, обязательно выделять элементарные высказывания и правильно расставлять скобки, определяя порядок действий. При изучении свойств операций (коммутативность, ассоциативность, дистрибутивность) следует использовать параллели с привычными арифметическими действиями, а для специфических законов (де Моргана, поглощения) применять опорные карточки и таблицы эквивалентных преобразований. При построении таблиц истинности необходимо требовать обязательного выделения промежуточных столбцов, соответствующих каждой логической операции, что формирует навык пошагового выполнения действий и предотвращает пропуск преобразований.

6.Начальное программирование как средство развития алгоритмического мышления. Несмотря на то, что задания высокого уровня сложности остаются для данной группы малодоступными, начинать систематическую работу по формированию навыков программирования необходимо с 8 класса. Последовательность усложнения должна быть следующей: линейные алгоритмы, условный оператор (поиск максимума, проверка чётности), цикл со счётчиком (вывод последовательности), обработка цифр натурального числа. Каждый следующий тип задач должен опираться на ранее освоенные конструкции.

Обязательным требованием является тестирование программы на нескольких наборах входных данных, включая граничные случаи (нулевые значения, отрицательные числа, минимальные и максимальные значения из допустимого диапазона). На начальном этапе учитель предоставляет готовые наборы тестов, затем учащиеся учатся подбирать их самостоятельно. Это формирует навык самопроверки и позволяет выявлять логические ошибки, которые не видны при тестировании только на примере из условия задачи. Эффективным приёмом является запись ключевых конструкций (поиск суммы цифр, проверка простоты). Обязательным требованием является тестирование программы на нескольких наборах входных данных, включая граничные случаи (нулевые значения, отрицательные числа, минимальные и максимальные значения из допустимого диапазона). На начальном этапе учитель предоставляет готовые наборы тестов, затем учащиеся учатся подбирать их самостоятельно. Это формирует навык самопроверки и позволяет выявлять логические ошибки, которые не видны при тестировании только на примере из условия задачи.

Также рекомендуется практиковать устное проговаривание алгоритма перед написанием кода: учащийся должен объяснить, что именно и в какой последовательности будет делать программа, какие переменные потребуются и

как они будут меняться. Это развивает регулятивные УУД и предотвращает ошибки, связанные с поспешным написанием кода без предварительного планирования.

Для отработки синтаксиса полезны короткие задания на исправление ошибок в готовом коде (целенаправленно внесённые синтаксические или логические ошибки).

7.Формирование умения перехода от условия задачи к программной реализации. Для развития данного умения целесообразно внедрить приём письменной фиксации четырёх этапов решения: анализ условия и выделение исходных данных; построение формальной модели задачи; разработка алгоритма решения (словесно или в виде блок-схемы); написание программы на языке программирования. Данный приём структурирует мыслительную деятельность, предотвращает пропуск важных этапов и развивает регулятивные универсальные учебные действия. На начальном этапе учитель демонстрирует каждый этап на примере, затем учащиеся выполняют их самостоятельно, а на заключительном этапе переходят к краткой фиксации только ключевых моментов.

8.Целенаправленная отработка арифметики в позиционных системах счисления. В процессе обучения необходимо формировать навык свободного оперирования числами в различных системах счисления и выполнения арифметических операций. Рекомендуется выделять 3–5 минут на устные и письменные упражнения по переводу чисел между системами счисления (десятичной, двоичной, восьмеричной, шестнадцатеричной) в диапазоне 0–255, а также на выполнение сложения и умножения в недесятичных системах. Эффективным приёмом является использование таблиц степеней оснований систем счисления, которые учащиеся заполняют самостоятельно и используют как опору при решении. На начальном этапе важно требовать записи развёрнутой формы числа и проверки обратным переводом.

9.Развитие метапредметных умений. Формирование предметных результатов у данной группы учащихся требует целенаправленного развития универсальных учебных действий. Приоритетными направлениями являются смысловое чтение, которое предполагает на каждом уроке практиковать разбор условия задачи с выделением ключевых слов, определением исходных данных и требуемого результата, обсуждением возможных способов решения; самоконтроль и рефлексия через использование листов самооценки, критериальных рубрик, взаимопроверки, а также требование письменного или устного комментирования выполненных действий; планирование, реализуемое через составление краткого плана в словесной форме или в виде блок-схемы перед выполнением сложного задания.

Реализация данных рекомендаций должна начинаться не в выпускном классе, а по мере выявления дефицитов — начиная с основной школы. Только системная и последовательная работа, опирающаяся на актуальные методические разработки и анализ реальных ошибок, позволит обучающимся с низким уровнем подготовки не только успешно преодолеть минимальный порог ЕГЭ, но и сформировать целостное представление о предмете и его методах.

Рекомендации для учителей по совершенствованию преподавания учебного предмета группе обучающихся со средним уровнем подготовки

Настоящие рекомендации адресованы учителям, работающим с обучающимися, чей уровень подготовки может быть охарактеризован как средний: они уверенно выполняют задания базового и части повышенного уровней сложности, но испытывают затруднения при решении задач, требующих многошаговых рассуждений и выбора нестандартного подхода. Для данной группы приоритетом становится систематизация и углубление имеющихся умений, развитие навыков самостоятельного планирования, самоконтроля и рефлексивной оценки собственных решений. Системный подход к преподаванию на этом этапе предполагает переход от решения изолированных задач к работе с комплексными проблемами, интегрирующими различные темы и методы.

1. Организация практикумов по программированию обработки символьных данных. Для формирования устойчивого навыка работы со строковыми типами данных рекомендуется организовать серию практикумов, на которых учащиеся последовательно осваивают следующие типы задач: подсчёт количества появлений заданного символа в строке, разбиение строки на слова по разделителю, поиск и замена подстрок, извлечение подстроки по заданным условиям, обработка строк, содержащих цифры и буквы одновременно. На начальном этапе учащиеся работают с готовыми алгоритмами, анализируя их и модифицируя под изменённые условия, затем переходят к самостоятельной реализации. Особое внимание следует уделять обработке краевых случаев: пустая строка, строка из одного символа, строка без разделителей, строка, состоящая только из разделителей. Каждое практическое занятие должно завершаться тестированием программы на наборе из 3–5 контрольных примеров, включая граничные значения.

2. Изучение и реализация алгоритмов сортировки и обработки массивов. Для формирования умения обрабатывать структурированные данные и находить оптимальные решения необходимо освоить базовые алгоритмы сортировки и методы анализа отсортированных данных. Рекомендуется поэтапное изучение: сначала реализация простых алгоритмов сортировки (пузырьком, вставками) на небольших массивах с визуализацией процесса, затем переход к более эффективным методам (сортировка подсчётом, быстрая сортировка) и сравнению их временной сложности. После освоения сортировки следует переходить к задачам, требующим однократного просмотра отсортированного массива для нахождения оптимального решения (например, выбор пар элементов с минимальной разностью, распределение ресурсов). На каждом занятии полезно использовать задания с изменёнными условиями, требующие адаптации уже изученных алгоритмов к новой постановке задачи.

3. Организация практикумов по анализу данных. Для формирования умения обрабатывать большие объёмы данных и выявлять закономерности рекомендуется организовать серию практикумов, на которых учащиеся выполняют полный цикл обработки данных: сбор и первичная обработка, очистка и подготовка данных, построение модели, интерпретация результатов. Учащиеся

работают с реальными или максимально приближенными к реальным наборами данных — таблицами, текстовыми файлами, результатами измерений. На каждом занятии учащиеся выполняют полную последовательность действий: загрузка данных из файла, предварительная обработка (фильтрация, удаление пропусков), выбор оптимальной структуры данных для хранения, реализация алгоритма обработки, вывод и интерпретация результатов. Особое внимание следует уделять анализу полученных результатов: учащийся должен не только получить числовой ответ, но и объяснить, что означает полученное значение в контексте условия задачи, проверить его на правдоподобность и сопоставить с ожидаемыми закономерностями.

4. *Формирование навыка полного самоконтроля при программировании.* Для успешного выполнения заданий, связанных с разработкой программ, необходимо сформировать у обучающихся навык системного тестирования разрабатываемых программ. Рекомендуется ввести обязательное требование проверки программы на граничных случаях: минимальные и максимальные значения входных данных, пустые последовательности, строки минимальной и максимальной длины, нулевые и отрицательные значения, большие объёмы данных (для оценки производительности). На начальном этапе учитель предоставляет готовые наборы тестов и демонстрирует, как их анализ помогает выявить скрытые ошибки. Затем учащиеся учатся самостоятельно подбирать тесты, ориентируясь на ограничения, указанные в условии задачи, и на возможные «слабые места» реализованного алгоритма. Полезно практиковать взаимопроверку программ, когда один учащийся подбирает тесты для программы, написанной другим, что развивает критическое мышление и умение находить ошибки в чужом коде. Эффективным приёмом является ведение «журнала отладки», в котором фиксируются обнаруженные ошибки и способы их устранения, что способствует накоплению опыта и повышению качества самопроверки в будущем.

5. *Формирование навыка отладки программ.* Важно не только проверять программу на разных данных, но и учить ребят находить ошибки с помощью отладчика — инструмента, который есть в любой среде программирования. На каждом практикуме выделяйте 5–7 минут на разбор типичных ошибок и их поиск в готовом коде. Можно давать специальные задания: найти и исправить ошибки в уже написанной программе. Это помогает ученикам видеть свои ошибки, не тратить время на бесконечный поиск и писать более качественный код.

6. *Внедрение проектных заданий, интегрирующих несколько тем.* Для перехода от изолированных задач к комплексным проблемам рекомендуется предлагать учащимся проектные задания, требующие объединения навыков работы с файлами, обработки массивов, сортировки, поиска и анализа данных. Например, разработка программы для анализа успеваемости класса с выводом статистических отчётов, или создание системы рекомендаций на основе данных. Такие задания развивают умение планировать работу, распределять этапы, тестировать готовое решение и представлять результаты. Проекты могут выполняться индивидуально или в парах, что дополнительно развивает коммуникативные универсальные учебные действия.

7. *Осуществление системной работы над метапредметными умениями:* смысловым чтением условия задачи, декомпозицией задачи на подзадачи, планированием последовательности действий, рефлексивной оценкой собственных решений и тестированием на граничных случаях. Именно эти умения в значительной степени определяют успешность выполнения заданий повышенного и высокого уровней сложности, а также способность самостоятельно переносить освоенные алгоритмические конструкции в новые контексты.

Реализация предложенных рекомендаций позволит педагогу выстроить системную работу по углублению предметных знаний и развитию алгоритмического мышления у учащихся, переходя от решения изолированных задач к комплексным проблемам, интегрирующим различные темы и методы.

Рекомендации для учителей по совершенствованию преподавания учебного предмета группе обучающихся с высоким уровнем подготовки

1. *Развитие навыков эффективного программирования.* Для углублённой подготовки по программированию целесообразно использовать задачи олимпиадного типа, требующие не только формальной реализации алгоритма, но и обоснованного выбора структуры данных с учётом ограничений по времени и памяти. При отборе заданий предпочтение следует отдавать задачам на обработку строк и массивов с требованием эффективности $O(n)$ или $O(n \log n)$, что непосредственно соотносится с критериями оценки заданий высокого уровня сложности. В процессе решения таких задач учащиеся осваивают навыки анализа входных данных, выбора оптимального алгоритма сортировки или однократного просмотра, а также учатся прогнозировать поведение программы на больших объёмах данных и граничных случаях, что формирует профессиональные компетенции, необходимые для достижения максимальных результатов. Рекомендуется включать такие задачи в систему регулярных практикумов, олимпиадных тренировок и домашних заданий, постепенно повышая их сложность и требование к эффективности.

2. *Моделирование параллельных вычислительных процессов.* Для углублённого изучения архитектуры ЭВМ и параллельных вычислений целесообразно включать в учебный процесс задачи, требующие построения временных диаграмм многопроцессорных систем с учётом зависимостей между процессами, что формирует у учащихся системное представление о распределённых вычислениях. При решении таких задач учащиеся должны анализировать логические связи между процессами, определять критические пути выполнения, оценивать время завершения при различном количестве процессоров и выявлять возможные простои оборудования, что развивает навыки формализации и интерпретации схем параллельного выполнения. Рекомендуется использовать визуализацию в виде диаграмм Ганта, таблиц загрузки процессоров и графов зависимостей, что позволяет перевести абстрактное описание алгоритма в наглядную модель, доступную для анализа. Систематическая работа с такими заданиями не только готовит к выполнению экзаменационных задач, но и

формирует фундаментальное понимание принципов работы современных вычислительных систем, включая многопоточность, синхронизацию и распределение ресурсов, что является важным компонентом ИКТ-компетенций на углублённом уровне.

3. Практикумы по комплексному анализу данных. Для развития умений работать с данными рекомендуется организовать практикумы, на которых ученики выполняют все этапы работы: загрузка данных из файла, их очистка и подготовка, выбор способа хранения, обработка с помощью электронных таблиц или языка программирования, построение графиков и формулировка выводов по полученным результатам. Важно, чтобы каждая задача проходила через всю цепочку — от исходных данных до итогового ответа с пояснениями. Такой подход учит выбирать удобный инструмент для каждой задачи, проверять правдоподобность результатов и обосновывать своё решение. В качестве заданий можно использовать реальные данные, например, погодные наблюдения или спортивные результаты.

4. Применение эффективных алгоритмов сортировки и поиска. Для успешного решения задач, связанных с обработкой больших массивов данных, необходимо научить учащихся применять эффективные алгоритмы сортировки (быстрая сортировка, сортировка слиянием) и поиска (бинарный поиск). На практике это означает, что ученик должен не просто знать теоретический материал, а уметь выбрать подходящий алгоритм в зависимости от объёма данных и условий задачи, а также использовать встроенные средства языка программирования. Рекомендуется включать задания, где требуется отсортировать массив по нескольким условиям или найти элемент за минимальное время, а также обсуждать, почему один алгоритм работает быстрее другого на больших объёмах. В качестве тренировки полезно давать задачи на обработку файлов с большим количеством строк, чтобы ученики на практике убедились в важности выбора эффективного алгоритма.

5. Формирование метапредметных умений. Для успешного решения задач высокого уровня сложности необходимо целенаправленно формировать у обучающихся регулятивные и познавательные универсальные учебные действия. Введите практику обязательного планирования перед написанием кода: учащийся должен составлять краткий план решения (словесно или в виде блок-схемы) с выделением подзадач и выбором структур данных. Используйте критериальные рубрики для оценивания, где учитываются не только правильность, но и эффективность алгоритма, читаемость кода и полнота тестирования, привлекайте школьников к самооценке и взаимопроверке.

6. Разнообразные форматы учебной деятельности. Для поддержки мотивации и углублённого изучения предмета рекомендуется использовать разнообразные формы работы. Учащимся можно предложить долгосрочные проекты, связанные с обработкой реальных данных, что интегрирует все изученные навыки и показывает практическую значимость предмета. Рекомендуйте школьникам участие в олимпиадах по программированию (ВсОШ, «Технокубок» и т. д.). Для особо подготовленных учащихся можно составлять

индивидуальные образовательные траектории с задачами повышенной сложности и самостоятельной работой на онлайн-платформах.

Рекомендуемые темы для обсуждения/ обмена опытом на школьных методических объединениях учителей-предметников, в том числе по трансляции эффективных педагогических практик ОО со стабильно высокими результатами ЕГЭ

На основе анализа типичных ошибок участников ЕГЭ по информатике 2026 года и выявленных предметных дефицитов для рассмотрения на заседаниях школьных методических объединений рекомендуется включить следующие темы:

1. Формирование базовых алгоритмических умений в 8–9 классах как содержательная основа преемственности при подготовке к государственной итоговой аттестации по информатике.
2. Организация практикумов по работе с электронными таблицами: от освоения базовых функций к решению комплексных задач анализа данных.
3. Методика систематического обучения программированию обработки строк и массивов в 10–11 классах как условие формирования алгоритмической компетентности.
4. Развитие метапредметных умений формализации условий задачи и самоконтроля в процессе обучения информатике.
5. Методические подходы к подготовке обучающихся к выполнению заданий высокого уровня сложности: освоение эффективных алгоритмов сортировки и комплексного анализа данных.
6. Формирование умений работы с позиционными системами счисления на основе алгоритмического подхода.
7. Методика обучения решению задач на игровые стратегии и построение дерева перебора.
8. Использование цифровых образовательных ресурсов для подготовки к ЕГЭ по информатике.
9. Преемственность в обучении информатике между основной и старшей школой как условие успешной подготовки к ГИА.
10. Развитие регулятивных универсальных учебных действий при подготовке к экзамену: планирование, самоконтроль и рефлексия.

Рекомендуемые ИПК / ИРО, иным организациям, реализующим программы профессионального развития учителей, направления повышения квалификации(в форме курсов, отдельных мероприятий, консультаций и т.п.) с учетом выявленных дефицитов предметной подготовки

Сформировать систему повышения квалификации учителей информатики, направленную на совершенствование преподавания учебного предмета с учетом

основных проблем обучения, результатов освоения содержательных разделов курса информатики (для разного уровня сложности), дефицитов образовательных результатов обучающихся и рекомендаций по их ликвидации.

1. Тематики программ обучения:

- Методика преподавания темы «Электронные таблицы» в контексте требований ФГОС СОО и ФОП: практикумы по анализу данных.
- Современные методические подходы к обучению программированию в школе: от освоения базовых алгоритмических конструкций к обработке строк и массивов и применению эффективных алгоритмов сортировки.
- Формирование метапредметных умений формализации условий задачи, моделирования и самоконтроля в процессе обучения информатике.
- Методика подготовки обучающихся к выполнению заданий на анализ алгоритмов и построение игровых стратегий.
- Методические аспекты использования открытых форматов файлов (.ods, .odt, .txt) в процессе подготовки к государственной итоговой аттестации по информатике.
- Методика преподавания раздела «Архитектура ЭВМ и параллельные вычисления»: формирование умений анализа зависимостей и построения временных диаграмм.
- Методика преподавания раздела «Компьютерные сети. Адресация в сети Интернет. Маска подсети»: формирование умений работы с IP-адресацией и поразрядной конъюнкцией.

2. Осуществлять адресное информационно-методическое сопровождение учителей информатики образовательных организаций, в которых учащиеся получили низкие результаты ЕГЭ по предмету, а также молодых и вновь устроенных специалистов (учителей информатики).

3. Продолжить работу по созданию условий для повышения уровня мастерства учителей информатики через участие в мастер-классах, круглых столах, семинарах, а также через организацию системы работы по самообразованию и обмену педагогическим опытом.

Рекомендации по другим направлениям (при наличии)

1. Муниципальным органам управления образования и методическим службам:

- провести анализ кадрового обеспечения преподавания информатики в школах муниципалитета, обратив внимание на профессиональные компетенции педагогов, работающих в 10–11 классах, поскольку на этом этапе формируются углублённые навыки программирования и анализа данных, дефицит которых проявляется на ЕГЭ;

- организовать муниципальные семинары-практикумы по обмену опытом между школами с высокими и низкими образовательными результатами, уделяя особое внимание методике обучения программированию, обработке данных и подготовке к заданиям высокого уровня сложности (24–27);
- содействовать развитию системы кружков и элективных курсов по программированию и анализу данных в 8–11 классах для подготовки к заданиям высокого уровня сложности;

2. Администрациям образовательных организаций:

- обеспечить образовательные организации лицензионным или свободным программным обеспечением (редакторы электронных таблиц, текстовые редакторы, среды программирования на языках C#, C++, Pascal, Java, Python) в соответствии с перечнем, утверждённым для проведения ЕГЭ по информатике;
- организовать работу по раннему выявлению школьников, планирующих сдавать ЕГЭ по информатике (информирование законных представителей о важности и процедуре проведения государственной итоговой аттестации, анкетирование учащихся по выбору предметов);
- обеспечить проведение диагностики предметных и метапредметных знаний с целью дифференциации учащихся по уровню подготовки;
- обеспечить возможность проведения дополнительных занятий по подготовке к ЕГЭ и индивидуальных консультаций;
- обеспечить регулярное информирование родителей о промежуточных результатах и проблемах школьников при подготовке к ЕГЭ;
- скорректировать систему внутришкольного мониторинга и контроля подготовки к ЕГЭ по информатике для учащихся разного уровня предметной подготовки;
- обеспечить возможность применения цифровых образовательных материалов для организации дифференцированного обучения школьников;
- организовать мониторинг качества преподавания информатики в образовательной организации с последующей адресной методической поддержкой учителей, чьи учащиеся демонстрируют стабильно низкие результаты.