



РЕКОМЕНДАЦИИ УЧИТЕЛЯМ ПО СОВЕРШЕНСТВОВАНИЮ ПРЕПОДАВАНИЯ УЧЕБНОГО ПРЕДМЕТА «ИНФОРМАТИКА» ДЛЯ ВСЕХ ОБУЧАЮЩИХСЯ ГОРОДА МОСКВЫ

Группе обучающихся с минимальным уровнем подготовки

На основе выявленных дефицитов в предметной подготовке и недостаточной сформированности метапредметных результатов ФГОС у участников, не преодолевших минимальный балл, рекомендуется следующее:

1. Устранение пробелов в вычислительных навыках и работе с числами (начинать с первого года изучения информатики и систематически продолжать в последующих классах)

Участники данной группы испытывают затруднения с выполнением арифметических операций и переводом единиц измерения, что напрямую влияет на успешность выполнения заданий на измерение количества информации и кодирование.

- Методика «Информационный калькулятор». Включить в уроки информатики (5–10 минут) задания на перевод единиц измерения информации (бит, байт, Кбайт, Мбайт). Использовать карточки с разноуровневыми заданиями: от простого перевода величин до вычисления объема файла по формуле, без использования электронных таблиц на первых этапах.

- Прием «Проверь соседа». Парная работа, где ученики проверяют друг у друга правильность перевода единиц и вычислений по формулам (например, формула Хартли или определение объема звукового файла), обсуждают ошибки и объясняют правильный способ действия. Это развивает и предметные, и коммуникативные УУД.

2. Выравнивание уровня подготовки в решении текстовых задач (формализация алгоритмов) (начинать с первого года изучения информатики)

Неумение переводить текст на естественном языке в формальный алгоритм или математическую модель является ключевым дефицитом, проявляющимся в заданиях на анализ алгоритмов и формальное исполнение.

- Методика «Алгоритм понимания задачи». Разработать и использовать единый алгоритм: «Читаем условие → выделяем данные и искомые величины → определяем связи между величинами → строим таблицу/схему → формализуем (записываем алгоритм или формулу) → выполняем → записываем ответ». Алгоритм



должен быть представлен в виде памятки и отработан на большом количестве примеров.

- Прием «Структурирование условия». При решении задач на исполнение алгоритмов для учебных исполнителей (Робот, Черепаха) учить детей маркировать ключевые параметры в тексте: «Дано», «Что нужно получить», «Последовательность команд». Обводить важные числа, подчеркивать условия окончания цикла.

- Прием «Переформулируй своими словами». Учить пересказывать условие задачи на естественном языке, заменяя сложные обороты простыми. Это помогает выделить суть задачи и снять страх перед «сложными» формулировками.

3. Повышение уровня в работе с информационными моделями (схемы, таблицы, графы) (начинать с 9 класса)

Несформированность навыков анализа и структурирования данных проявляется в трудностях при работе с таблицами истинности, графами и реляционными базами данных.

- Создание «Карточки-шпаргалки». Вместе с учениками составить и выучить мини-справочник по основным элементам таблицы истинности (конъюнкция, дизъюнкция, отрицание, импликация) и способам кодирования (условие Фано). Карточка должна быть всегда доступна на начальном этапе, затем постепенно убираться.

- Методика «Моделирование в таблицах». Предлагать задания на заполнение готовых таблиц, где часть данных уже вписана, а часть нужно рассчитать. Это снижает порог входа и учит видеть логические связи. Постепенно переходить к самостоятельному построению таблиц для произвольных логических функций.

- Прием «Сворачивание и разворачивание». Учить строить дерево (например, для кодирования по Фано или для выигрышных стратегий в играх), а затем обратно, восстанавливать условие по дереву. Отрабатывать параллельно, используя простые примеры с небольшим количеством элементов.

4. Повышение сформированности компетенций в преобразовании логических выражений и алгоритмов (начинать с 8 класса)

Затруднения при выполнении преобразований логических выражений и формализации условий являются следствием низкого уровня абстрактного мышления.

- Систематическая работа с логическими операциями. Выделить 3–4 ключевых логических правила (законы де Моргана, распределительный закон) и



отрабатывать их применение на тренажерах. Тренажеры должны быть короткими (5–7 минут), но регулярными.

- Методика «Найди ошибку в алгоритме». Предлагать готовые алгоритмы (на естественном языке или с помощью блок-схем) с типичными ошибками (неправильная последовательность действий, нарушение условия цикла). Задание: найти и исправить ошибку. Это развивает внимание, анализ и самоконтроль (регулятивные УУД).

- Прием «Похожие, но разные». Предлагать несколько логических выражений, которые выглядят похоже, но требуют применения разных законов преобразования. Учить распознавать структуру выражения.

5. Формирование регулятивных УУД (самоконтроль и самоорганизация) (на протяжении всего курса изучения информатики)

Регулятивные дефициты (неумение спланировать решение и проверить результат) являются системными для данной группы и требуют длительной коррекции.

- Обучение самопроверке. Выработать универсальный алгоритм проверки решения для любых задач: «Проверь, все ли условия учтены → проверь правильность переноса чисел в вычислениях → подставь полученный ответ (результат) в условие задачи → оцени, имеет ли полученный ответ смысл». Систематически использовать этот алгоритм при выполнении любых заданий в классе.

- Стратегия «Сначала простые задания». Обучить учеников на экзамене (и на контрольных работах) начинать с заданий, в которых они уверены, чтобы гарантированно получить баллы, и только потом переходить к более сложным, вызывающим сомнения. Отрабатывать эту стратегию на тренировочных работах.

- Управление временем. Научить учеников быстро оценивать сложность задания и не тратить более 5–7 минут на одно задание, если решение не получается. Научить «откладывать» трудное задание и возвращаться к нему позже. Использовать таймер на тренировочных работах.

- Прием «Планирование работы». Учить составлять план решения задачи перед выполнением: «Что я знаю? Что нужно найти? Какие формулы или алгоритмы можно применить? В каком порядке это делать?».

6. Формирование познавательных УУД (анализ, синтез, моделирование) (на протяжении всего курса изучения информатики)



Познавательные дефициты (неумение структурировать информацию и строить модели) напрямую связаны с низкой успешностью выполнения заданий на поиск информации и анализ баз данных.

- Прием «Чтение с пометками». При решении любой задачи учить маркировать данные: «Дано», «Найти», «Известно», «Неизвестно». Это развивает умение структурировать информацию и отделять важное от второстепенного.

- Работа со схемами и опорами. Использовать визуализацию (графы, таблицы, кластеры, блок-схемы) для систематизации информации, особенно при решении задач на анализ баз данных и построение логических игр. Учить переводить текст в схему и обратно.

- Прием «От частного к общему». Учить на примере нескольких конкретных задач выявлять общие закономерности (например, правило построения кодов по условию Фано), делать выводы и формулировать правила.

7. Общие методические рекомендации по работе с группой

- Индивидуализация. Рекомендуется разработать для каждого ученика из данной группы индивидуальный маршрут коррекции с учетом выявленных конкретных дефицитов (например, кто-то «плавает» в степенях двойки, а кто-то не понимает условия Фано).

- Систематичность. Работа должна быть регулярной (не от случая к случаю), с постепенным наращиванием сложности. Важно выделить 3–4 ключевые темы (например, кодирование информации, построение таблиц истинности, анализ простых алгоритмов) и доводить их до автоматизма, не распыляясь на всё сразу.

- Зона ближайшего развития. Предлагать задания, которые ученик может решить с небольшой помощью учителя или одноклассника, постепенно снижая уровень поддержки.

- Позитивное подкрепление. Крайне важно фиксировать даже небольшие успехи и прогресс, формируя у ученика уверенность в своих силах. Критика должна быть конструктивной и направленной на конкретные ошибки.

Представленные методики и приемы носят системный характер и должны реализовываться в образовательном процессе, начиная с первого года изучения информатики, а не только на этапе интенсивной подготовки к ЕГЭ в 10–11 классах.

Работа с группой обучающихся, рискующих не преодолеть минимальный балл, должна быть системной, последовательной и терпеливой. Основная задача на начальном этапе – не «дать максимум знаний», а сформировать базовые навыки и



обеспечить преодоление минимального порога. Успех достигается за счет: регулярной отработки базовых умений (вычисления, простые логические операции); формирования алгоритмического и структурного мышления; развития навыков самоконтроля и планирования; создания ситуации успеха и поддержки мотивации. Представленные рекомендации адресованы учителям информатики всех классов и должны реализовываться в системе на протяжении всего периода обучения, а не только в выпускных классах.

Вывод:

Участники, не преодолевшие минимальный порог, демонстрируют фрагментарные предметные знания, сконцентрированные в основном на заданиях, требующих прямого считывания информации из таблиц, графов и баз данных. Их системные дефициты связаны с невладением базовыми вычислительными навыками (перевод единиц, степени двойки), непониманием формальных алгоритмов и отсутствием навыков смыслового чтения. Критическим является низкий процент выполнения заданий на обработку числовой информации в электронных таблицах (3%) и на логические игры (менее 10%). Метапредметные дефициты (неумение планировать, контролировать промежуточные результаты, интерпретировать текстовые условия) усугубляют предметные пробелы. Реалистичная цель для данной группы – преодоление минимального порога (40 т.б.) за счёт целенаправленной отработки 5–6 ключевых тем (кодирование, таблицы истинности, простые алгоритмы) и формирования элементарных регулятивных навыков (проверка ответа, работа с памятками). Работа по устранению существующих пробелов в усвоении информатики данной группой обучающихся должна носить системный, пошаговый характер и начинаться с 7–8 класса, а не только на этапе подготовки к ЕГЭ.



Группе обучающихся с низким уровнем подготовки

На основе выявленных дефицитов в предметной подготовке и недостаточной сформированности метапредметных результатов ФГОС у участников, набравших 40–60 баллов, рекомендуется следующее:

1. Устранение выявленных пробелов в работе с электронными таблицами и анализом данных (начинать с 9 класса, систематически продолжать в 10–11 классах)

Участники данной группы испытывают затруднения с формализацией задач для электронных таблиц. Рекомендуется сконцентрироваться на практико-ориентированных задачах.

- Методика «Пошаговое составление формул». Учить разбивать сложное условие на несколько простых логических частей (например, «найти сумму чисел, кратных 5 и находящихся в диапазоне A1:A10»). Отрабатывать алгоритм: «Определи диапазон → Определи условие → Выбери функцию → Запиши аргументы».

- Прием «Расшифруй формулу». Предлагать учащимся готовые электронные таблицы с формулами. Задание: описать словами, какие действия выполняет формула (например, «суммирует значения в столбце В, если в столбце А стоит текст "Да"»). Это развивает умение читать и понимать язык формул (познавательные УУД).

- Прием «Найди ошибку в формуле». Предлагать готовые решения с типичными ошибками (неверные ссылки, ошибки в условиях). Это развивает внимание и самоконтроль (регулятивные УУД).

2. Реализация мероприятий по устранению отставаний в решении задач на игровые стратегии (начинать с 10–11 классов)

Непонимание логики игровых стратегий является следствием несформированности умений анализировать и строить деревья решений.

- Методика «Дерево решений». На начальном этапе предлагать учащимся готовые шаблоны деревьев игр, где часть ветвей уже построена, а часть нужно дописать. Постепенно переходить к самостоятельному построению.

- Прием «Анализ ситуации». Учить выделять в условии ключевые слова: «неудачный ход Пети» (это условие, при котором Ваня выигрывает), «выигрышная стратегия Пети» (это условие, при котором Петя гарантирует себе победу). Отрабатывать четкое разграничение этих понятий.

- Прием «Визуализация». Использовать цветные таблицы и схемы для обозначения выигрышных и проигрышных позиций (например, красный – проигрыш,



зеленый – выигрыш). Это помогает структурировать информацию (познавательные УУД).

3. Минимизация разрывов в работе с массивами и последовательностями чисел (начинать с 9 класса)

- Методика «Пошаговый перебор». Учить описывать на естественном языке, как работает алгоритм поиска элемента в массиве. Предлагать задачи с однократным просмотром массива (например, «найти минимальный элемент», «подсчитать количество четных чисел»).
- Прием «Анализ изменения массива». Предлагать готовые алгоритмы обработки массива и просить учащихся отследить, как меняются значения элементов на каждом шаге. Заполнять таблицу состояний. Это развивает навыки анализа и контроля (регулятивные УУД).

4. Восполнение дефицитов в преобразовании и понимании информации (измерение количества, кодирование) (начинать с 8–9 классов)

- Систематическая работа с единицами измерения. Составить памятку «Степени двойки» и «Перевод единиц» и отрабатывать их на коротких (5–7 минут) тренажерах в начале урока.
- Методика «Алфавитный подход». Отрабатывать на простых примерах (определение объема сообщения, если известна мощность алфавита). Учить применять формулу Хартли для равновероятных событий.
- Прием «Связь с жизнью». Предлагать задачи, связанные с реальными ситуациями (например, объем цифровой фотографии, звукового файла). Это повышает мотивацию и позволяет лучше понять суть задач.

5. Формирование регулятивных УУД (самоконтроль и самоорганизация) (на протяжении всего периода изучения информатики)

Регулятивные дефициты (неумение спланировать решение и проверить результат) являются системными для данной группы и требуют длительной коррекции.

- Обучение самопроверке. Выработать универсальный алгоритм проверки решения: «Проверь, все ли условия учтены → проверь правильность переноса чисел в вычисления → подставь полученный ответ (результат) в условие задачи → оцени, имеет ли полученный ответ смысл».
- Стратегия «Сначала простые задания». Обучить учеников на экзамене (и на контрольных работах) начинать с заданий, в которых они уверены (базовый



уровень), и только потом переходить к более сложным. Отрабатывать эту стратегию на тренировочных работах.

- Управление временем. Научить учеников быстро оценивать сложность задания и не тратить более 5–7 минут на одно задание, если решение не получается. Использовать таймер на тренировочных работах.
- Прием «Планирование работы». Учить составлять план решения задачи перед выполнением: «Что я знаю? Что нужно найти? Какие формулы или алгоритмы можно применить? В каком порядке это делать?».

6. Формирование познавательных УУД (анализ, синтез, моделирование) (на протяжении всего периода изучения информатики)

- Прием «Чтение с пометками». При решении любой задачи учить маркировать данные: «Дано», «Найти», «Известно», «Неизвестно». Это развивает умение структурировать информацию.
- Работа со схемами и опорами. Использовать визуализацию (таблицы, кластеры, блок-схемы) для систематизации информации, особенно при решении задач на анализ игр и баз данных.
- Прием «От частного к общему». Учить на примере нескольких конкретных задач выявлять общие закономерности (например, алгоритм нахождения выигрышной позиции), делать выводы и формулировать правила.

7. Общие методические рекомендации по работе с группой

- Индивидуализация. Рекомендуется разработать для каждого ученика из данной группы индивидуальный маршрут коррекции с учетом выявленных конкретных дефицитов (например, кто-то «плавает» в электронных таблицах, а кто-то не понимает игровые стратегии).
- Систематичность. Работа должна быть регулярной (не от случая к случаю), с постепенным наращиванием сложности. Важно выделить 2–3 ключевые темы (например, обработка данных в электронных таблицах, игровые стратегии) и доводить их до автоматизма.
- Зона ближайшего развития. Предлагать задания, которые ученик может решить с небольшой помощью учителя или одноклассника, постепенно снижая уровень поддержки.
- Позитивное подкрепление. Крайне важно фиксировать даже небольшие успехи и прогресс, формируя у ученика уверенность в своих силах.



Представленные методики и приемы носят системный характер и должны реализовываться в образовательном процессе начиная с основной школы (7–9 классы), а не только на этапе интенсивной подготовки к ЕГЭ в 10–11 классах.

Работа с группой обучающихся с низким уровнем подготовки (40–60 баллов) должна быть системной, последовательной и направленной на преодоление порога в 60–70 баллов. Основная задача на начальном этапе – сформировать базовые навыки работы с информацией и алгоритмами. Успех достигается за счет: регулярной отработки базовых умений (работа с электронными таблицами, массивами); формирования структурного мышления (игровые стратегии, деревья); развития навыков самоконтроля и планирования. Представленные рекомендации адресованы учителям информатики всех классов и должны реализовываться в системе, а не только в выпускных классах.

Вывод:

Группа участников, набравших от 40 до 60 баллов, владеет базовыми операциями (построение таблиц истинности, поиск в базах данных), но испытывает серьёзные затруднения при переходе от репродуктивных действий к аналитическим. Основные дефициты сосредоточены в темах: обработка данных в электронных таблицах (процент выполнения 44%), игровые стратегии (26%), многопроцессорные системы (23%) и массивы (менее 10%). Типичные ошибки связаны с неумением формализовать условия в виде формул или алгоритмов, неспособностью отслеживать изменение состояния объекта (массива, дерева игры) и недостаточным контролем промежуточных результатов. Для перехода в группу 60–70 баллов необходимо сконцентрироваться на отработке типовых алгоритмов работы с электронными таблицами (суммы, поиск, логические функции) и построении деревьев игр для простых случаев. Метапредметные ориентиры: развитие смыслового чтения (выделение ключевых условий) и планирования последовательности шагов. Рекомендуется систематическая работа с постепенным усложнением задач.



Группе обучающихся со средним уровнем подготовки

На основе выявленных дефицитов в предметной подготовке и недостаточной сформированности метапредметных результатов ФГОС у участников, набравших 61–80 баллов, рекомендуется следующее.

1. Выявление проблемных зон в работе с массивами и последовательностями чисел (начинать с 9 класса, систематически продолжать в 10–11 классах)

Участники данной группы испытывают затруднения с отслеживанием изменения состояния массива в процессе выполнения алгоритма. Рекомендуется сконцентрироваться на задачах с однократным просмотром массива.

- Методика «Пошаговое выполнение алгоритма». Учить описывать на естественном языке, как работает алгоритм обработки массива на каждом шаге. Предлагать задачи, требующие отслеживания изменения состояния массива (например, «найти максимальный элемент», «подсчитать количество чисел, удовлетворяющих условию»).

- Прием «Анализ изменения состояния». Предлагать готовые алгоритмы обработки массива и просить учащихся отследить, как меняются значения элементов на каждом шаге. Заполнять таблицу состояний. Это развивает навыки анализа и контроля (регулятивные УУД).

- Прием «Алгоритмизация». Учить составлять алгоритм обработки последовательности чисел, используя базовые конструкции (цикл, условие). Отрабатывать на большом количестве примеров.

2. Устранение недостатков в решении задач на игровые стратегии (начинать с 10–11 классов)

Непонимание логики игровых стратегий в сложных условиях является следствием несформированности умений анализировать и строить деревья решений.

- Методика «Дерево решений с анализом условий». На начальном этапе предлагать учащимся готовые шаблоны деревьев игр, где часть ветвей уже построена, а часть нужно дописать. Постепенно переходить к самостоятельному построению с полным анализом условий («неудачный ход», «любой ход»).

- Прием «Анализ ситуации». Учить выделять в условии ключевые слова: «неудачный ход Пети» (это условие, при котором Ваня выигрывает), «выигрышная стратегия Пети» (это условие, при котором Петя гарантирует себе победу). Отрабатывать четкое разграничение этих понятий.



- Прием «Визуализация». Использовать цветные таблицы и схемы для обозначения выигрышных и проигрышных позиций (например, красный – проигрыш, зеленый – выигрыш). Это помогает структурировать информацию (познавательные УУД).

3. Преодоление дефицитов в работе с многопроцессорными системами и параллельными вычислениями (начинать с 10 класса)

Участники испытывают затруднения с пониманием логики параллельного выполнения и определением минимального времени выполнения серии процессов.

- Методика «Анализ зависимостей». Учить строить граф зависимостей процессов и определять, какие процессы можно выполнять параллельно, а какие – последовательно.

- Прием «Визуализация». Использовать цветные схемы для обозначения зависимостей (например, стрелки между процессами). Это помогает структурировать информацию (познавательные УУД).

- Прием «Алгоритмизация». Учить алгоритму определения минимального времени: «Определи зависимые процессы → Определи параллельные процессы → Вычисли время выполнения». Отрабатывать на большом количестве примеров.

4. Компенсация недостатков в анализе данных (кластеризация, классификация, анализ отклонений) (начинать с 10–11 классов)

Участники испытывают затруднения с пониманием сложных понятий и методов анализа данных. Рекомендуется сконцентрироваться на практико-ориентированных задачах.

- Методика «Анализ данных на примерах». Предлагать задачи, связанные с реальными ситуациями (например, классификация объектов, кластеризация точек на плоскости). Это повышает мотивацию и позволяет лучше понять суть задач.

- Прием «Визуализация». Использовать графические схемы для представления кластеров и результатов классификации. Это помогает структурировать информацию (познавательные УУД).

- Прием «Интерпретация». Учить описывать словами, что означают полученные результаты (например, «кластер – это группа объектов, близких друг к другу»). Это развивает навыки работы с текстом и интерпретации (коммуникативные УУД).

5. Формирование регулятивных УУД (самоконтроль и самоорганизация) (на протяжении всего периода изучения информатики)



Регулятивные дефициты (неумение спланировать решение и проверить результат) являются системными для данной группы и требуют длительной коррекции.

- Обучение самопроверке. Выработать универсальный алгоритм проверки решения: «Проверь, все ли условия учтены → проверь правильность переноса чисел в вычисления → подставь полученный ответ (результат) в условие задачи → оцени, имеет ли полученный ответ смысл».
- Стратегия «Сначала простые задания». Обучить учеников на экзамене (и на контрольных работах) начинать с заданий, в которых они уверены (базовый и повышенный уровень), и только потом переходить к сложным. Отрабатывать эту стратегию на тренировочных работах.
- Управление временем. Научить учеников быстро оценивать сложность задания и не тратить более 5–7 минут на одно задание, если решение не получается. Использовать таймер на тренировочных работах.
- Прием «Планирование работы». Учить составлять план решения задачи перед выполнением: «Что я знаю? Что нужно найти? Какие формулы или алгоритмы можно применить? В каком порядке это делать?».

6. Формирование познавательных УУД (анализ, синтез, моделирование) (на протяжении всего периода изучения информатики)

- Прием «Чтение с пометками». При решении любой задачи учить маркировать данные: «Дано», «Найти», «Известно», «Неизвестно». Это развивает умение структурировать информацию.
- Работа со схемами и опорами. Использовать визуализацию (таблицы, кластеры, блок-схемы) для систематизации информации, особенно при решении задач на анализ игр и баз данных.
- Прием «От частного к общему». Учить на примере нескольких конкретных задач выявлять общие закономерности (например, алгоритм нахождения выигрышной позиции, алгоритм определения минимального времени), делать выводы и формулировать правила.

7. Общие методические рекомендации по работе с группой

- Индивидуализация. Рекомендуется разработать для каждого ученика из данной группы индивидуальный маршрут коррекции с учетом выявленных конкретных дефицитов (например, кто-то «плавает» в игровых стратегиях, а кто-то не понимает параллельные вычисления).



- Систематичность. Работа должна быть регулярной (не от случая к случаю), с постепенным наращиванием сложности. Важно выделить 2–3 ключевые темы (например, обработка массивов и последовательностей, анализ данных, игровые стратегии) и доводить их до автоматизма.
- Зона ближайшего развития. Предлагать задания, которые ученик может решить с небольшой помощью учителя или одноклассника, постепенно снижая уровень поддержки.
- Позитивное подкрепление. Крайне важно фиксировать даже небольшие успехи и прогресс, формируя у ученика уверенность в своих силах.

Представленные методики и приемы носят системный характер и должны реализовываться в образовательном процессе, начиная с основной школы, а не только на этапе интенсивной подготовки к ЕГЭ в 10–11 классах. Работа с группой обучающихся со средним уровнем подготовки (61–80 баллов) должна быть системной, последовательной и направленной на преодоление порога в 80 баллов. Основная задача на начальном этапе – сформировать базовые навыки работы с массивами, игровыми стратегиями и анализом данных. Успех достигается за счет: регулярной отработки базовых умений (массивы, алгоритмы); формирования структурного мышления (анализ данных, многопроцессорные системы); развития навыков самоконтроля и планирования. Представленные рекомендации адресованы учителям информатики всех классов и должны реализовываться в системе, а не только в выпускных классах.

Вывод:

Участники со средним уровнем подготовки (61–80 баллов) демонстрируют уверенное владение заданиями повышенного уровня, связанными с вычислением рекуррентных выражений (91%), простыми игровыми стратегиями (76–77%) и алгоритмами для исполнителей (82%). Однако они «спотыкаются» на задачах, требующих комплексного анализа: неравномерное кодирование (успешность падает до 50–60%), параллельные вычисления (около 50 %) и работа с массивами с однократным просмотром (менее 50%). Типичные ошибки – неверное построение двоичного дерева для условия Фано, непонимание логики зависимостей в многопроцессорных системах и неумение отслеживать состояние массива при выполнении алгоритма. Для преодоления порога в 80 баллов необходимо



углублённо отработать алгоритмы сортировки, обработки массивов и решения задач на кластеризацию. Метапредметные акценты – развитие исследовательских действий (выявление закономерностей, проверка гипотез) и рефлексии (анализ собственных ошибок на граничных случаях). Работа должна вестись систематически в 10–11 классах на основе практико-ориентированных задач.



Группе обучающихся с высоким уровнем подготовки

На основе выявленных дефицитов в предметной подготовке и недостаточной сформированности метапредметных результатов ФГОС у участников, набравших 81–100 баллов, рекомендуется следующее.

1. Устранение дефицитов в алгоритмизации, оптимизации и обработке данных.

Участники данной группы хорошо владеют базовыми конструкциями языков программирования, однако испытывают затруднения в выборе оптимального по времени и памяти алгоритма, а также в корректной обработке нестандартных вводных данных.

- Методика «Оптимизация алгоритма». Предлагать учащимся одну и ту же задачу (например, на поиск простых чисел, делителей или сортировку) решить несколькими способами: от простого перебора до эффективных методов (решето Эратосфена, быстрая сортировка, алгоритмы с однократным просмотром массива). Проводить сравнение времени работы различных реализаций на больших тестовых наборах данных. Это помогает ученику осознанно подходить к выбору инструмента.

- Прием «Поиск неэффективного участка». Предлагать готовые программы, написанные формально верно, но с очевидной потерей производительности (например, вложенные циклы там, где можно обойтись однократным проходом). Задание: найти и исправить неэффективный участок, сократив время выполнения. Это развивает чувство алгоритмической сложности (познавательные УУД) и технику оптимизации кода.

- Методика «Обработка символьных строк и краевые эффекты». Научить и отработать обязательную проверку «граничных» условий при написании программ: пустая строка, пробел в начале и конце строки, одиночный символ-разделитель, несколько разделителей подряд. Учить использовать встроенные строковые функции в комбинации с ручным перебором для гибкости.

2. Преодоление пробелов в решении задач на построение моделей, анализ данных и параллельные вычисления.

Основные проблемы высокобалльников связаны с неверной интерпретацией связей в многопроцессорных системах и незавершенностью цикла анализа данных.

- Методика «Анализ графа зависимостей процессов». При решении задания №22 учить переводить текстовое описание ID процессов в наглядный граф. Отрабатывать алгоритм: «Определи все процессы с нулевой зависимостью →



выполни их параллельно → удали их из списка зависимостей для следующих процессов → повтори цикл».

- Прием «Синхронизация времени». При решении задач на параллельные системы учить учащихся не просто складывать время, а учитывать, что процесс может ждать завершения предыдущего. Отрабатывать на примерах, где несколько процессов ждут один и тот же завершающий процесс.

- Методика «Полный цикл анализа данных». Практиковать последовательность: «Сбор первичных данных → очистка (удаление выбросов) → оценка качества данных → выбор модели → визуализация → интерпретация результатов». Учить объяснять словами, что означают полученные числовые значения (например, «кластер – это группа близких объектов, а выброс – это аномалия»). Это развивает логику формализации и метапредметное умение переносить знания на практические ситуации.

3. Повышение уровня сформированности компетенций в работе с нестандартными условиями и комбинированными алгоритмами. Участники часто действуют по шаблону и ошибаются, когда в условии появляются неожиданные ограничения.

- Методика «Анализ изменения состояния». При решении задач на сортировку и перебор просить учеников отслеживать не просто финальный результат, а изменение состояния массива на каждом шаге выполнения алгоритма в уме или на черновике. Это развивает навык контроля (регулятивные УУД) и предотвращает ошибки логики при работе со сложными условиями поиска.

- Прием «Разбор нестандартных задач». Собрать банк заданий, где в условие были намеренно внесены «ловушки» (например, массивы с одинаковыми элементами, операции с числами вне диапазона, несколько условий сортировки одновременно). Решать такие задачи в классе, делая акцент на том, как именно нужно корректировать привычный алгоритм под измененные условия.

4. Формирование регулятивных УУД (самоконтроль, рефлексия, углубленная отладка).

Регулятивные дефициты (отсутствие проверки на краевые случаи, неумение отлаживать код) являются ключевым барьером для получения 100 баллов для этой группы.

- Обучение стратегии «Идеальная сдача». Учить учеников выделять на экзамене 10–15 минут на финальную проверку написанного кода. Ввести правило:



«Программа считается готовой, если она прошла тесты на всех граничных значениях, а не только на примере из условия». Отрабатывать стратегию тестирования собственных решений на тренировочных работах (использовать ручной ввод «крайних» значений).

- Стратегия «Рефлексия после решения». После решения сложной задачи просить ученика проговорить вслух (или записать): «Что я делал? Какие были ключевые моменты? Где я мог ошибиться в начале, если бы пошел не тем путем?». Это помогает выявить собственные «слепые зоны» в логике.

5. Формирование познавательных УУД (анализ, синтез, моделирование в нестандартных ситуациях).

Для высокобалльников типична ошибка, когда они «застревают» на привычном шаблоне и не видят более рационального пути решения.

- Прием «От частного к общему». Учить на примере 2–3 задач с разными формулировками (например, поиск максимального пути в матрице, разные условия обработки строк) выделять общий алгоритмический паттерн. Затем давать задачу с новым, незнакомым контекстом, но тем же паттерном, чтобы стимулировать перенос знаний.

- Работа со схемами и опорами. Для сложных задач на кластеризацию и классификацию требовать обязательного построения визуализации (набросок графика, таблицы или блок-схемы) до того, как приступить к написанию программы. Учить переводить словесное описание в визуальную модель и обратно.

6. Общие методические рекомендации по работе с группой.

- Отход от шаблонов. Учебный процесс для этой группы должен строиться не на заучивании типовых решений, а на развитии гибкости мышления и умения адаптировать алгоритмы под любые, даже самые нестандартные, формулировки.

- Индивидуальная траектория. Рекомендуется разрабатывать для каждого сильного ученика индивидуальный план, включающий в себя углубленное изучение тем, в которых он допускает редкие ошибки.

- Участие в олимпиадах. Поощрять участие высокобалльников в олимпиадах по информатике и математике, так как задачи на олимпиадах часто содержат те же «подводные камни», что и задания ЕГЭ, и развивают алгоритмическое мышление.



- **Позитивное подкрепление.** Важно фиксировать не только высокие результаты, но и прогресс в решении узких, ранее «провальных» для ученика тем, формируя у него уверенность в достижении максимального результата.

Представленные методики и приемы носят системный характер и должны реализовываться в процессе работы с мотивированными и одаренными учениками, начиная с 8–9 класса, с постепенным повышением уровня сложности. Работа с группой обучающихся с высоким уровнем подготовки (81–100 баллов) должна быть системной, последовательной и направленной на преодоление порога в 100 баллов. Успех достигается за счет: регулярной отработки нестандартных задач; углубленного изучения алгоритмов и структур данных; развития навыков рефлексии, самоконтроля и стратегии поведения на экзамене. Представленные рекомендации адресованы учителям информатики, работающим с профильными и олимпиадными классами, и должны реализовываться в системе, а не только на этапе финальной подготовки к ЕГЭ.

Вывод:

Высокобалльники (81–100 баллов) показывают стабильно высокие результаты по большинству заданий, включая игровые стратегии (96–97%), рекуррентные выражения (98%) и логические операции (95–96%). Основные потери баллов происходят на заданиях высокого уровня сложности: обработка символьных строк (№ 24 – 32%), целочисленных массивов с сортировкой (№ 26 – 38%) и многопроцессорных систем (№ 22 – 85%). Типичные ошибки носят «краевой» характер: неучёт граничных условий (пустые строки, пробелы, выход за пределы массива), неоптимальный выбор алгоритма (например, использование вложенных циклов вместо однократного просмотра) и недостаточная проверка на тестовых данных с нестандартными значениями. Для достижения 100 баллов ключевыми становятся развитие навыков рефлексии (осознанный выбор метода), самоконтроля (обязательное тестирование на «угловых» случаях) и углублённое изучение эффективных алгоритмов обработки данных. Рекомендуется вовлекать таких учеников в олимпиадное движение и предлагать задачи с нестандартными условиями, выходящими за рамки типовых шаблонов ЕГЭ.