

РЕКОМЕНДАЦИИ ДЛЯ СИСТЕМЫ ОБРАЗОВАНИЯ РОСТОВСКОЙ ОБЛАСТИ

Рекомендации по совершенствованию организации и методики преподавания предмета в Ростовской области на основе выявленных типичных затруднений и ошибок

ИНФОРМАТИКА

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

- **Формирование познавательных УУД (работа с информацией, логические операции, моделирование).**

1. *Развитие смыслового чтения и работы с текстовой информацией.* Учитывая, что значительная часть ошибок связана с непониманием условия задачи, необходимо систематически работать над развитием смыслового чтения на уроках информатики. Рекомендуемые методики и приёмы:

- Приём предтекстовая деятельность. На этом этапе важно актуализировать знания. Используйте приёмы вроде «Ассоциативного куста»: запишите на доске ключевое слово или заголовок и попросите учеников назвать все связанные с ним понятия и ассоциации. Это настраивает на работу с текстом. Также эффективны «Верные и неверные утверждения»: перед чтением ряда тезисов ученики оценивают их истинность, что активизирует мышление «Толстые и тонкие вопросы».

- Приём «Составление плана решения». Перед тем как приступить к решению, учащиеся должны составить план действий, записав его на черновике. Это помогает структурировать мысли и не пропускать важные шаги.

2. *Развитие логического мышления и работы с абстрактными моделями.* Для преодоления дефицитов, связанных с заданиями на логику и системы счисления, необходимо систематически развивать логическое мышление. Рекомендуемые методики и приёмы:

- Использование таблиц истинности в повседневной практике. На каждом уроке, где встречаются логические задачи, требовать от учащихся построения таблиц истинности для проверки своих решений. Начинать с простых выражений с двумя переменными, постепенно переходить к более сложным.

- Метод «Логических цепочек». При решении задач на логику (№ 2, № 15) требовать от учащихся записи полной цепочки рассуждений с пояснением каждого шага. Например, при упрощении выражения: «Используем закон де Моргана, получаем...», «Применяем закон поглощения...».

- Визуализация логических операций. Использовать цветовое кодирование для различных логических операций, схемы и диаграммы для визуализации логических связей. Создавать наглядные плакаты с таблицами истинности базовых операций.

- Организовать регулярные упражнения на перевод чисел из одной

системы счисления в другую, начиная с простых чисел и постепенно переходя к большим.

3. *Развитие навыков алгоритмического мышления.* Для преодоления дефицитов, связанных с заданиями на алгоритмизацию и программирование, необходимо развивать алгоритмическое мышление на всех этапах обучения. Рекомендуемые методики и приёмы:

- Использование учебных исполнителей. Для визуализации алгоритмов использовать среду учебного исполнителя (например, «КуМир», «Черепаша» в Python). Это позволяет учащимся «увидеть» результат выполнения алгоритма, связать абстрактные команды с конкретными действиями.

- Метод «Отладка программы». Давать учащимся программы с ошибками и просить найти их. Это развивает внимание, логическое мышление и понимание работы алгоритмов.

• **Формирование регулятивных УУД (планирование, самоконтроль, самооценка).**

1. *Развитие навыков планирования деятельности.* Учащиеся данной группы не умеют планировать свои действия, что приводит к беспорядочному решению задач и потере баллов. Рекомендуется использовать приём «План решения задачи». Перед решением каждой задачи требовать от учащихся составления письменного плана решения (на черновике). План должен содержать последовательность шагов и указание, что будет получено на каждом шаге. Использование алгоритмических предписаний. Для стандартных типов задач (например, для задач на кодирование или вычисление объёма памяти) разработать чёткие алгоритмические предписания, которые учащиеся должны выполнять пошагово.

2. *Развитие навыков самоконтроля и самопроверки.* Отсутствие навыков самоконтроля приводит к тому, что учащиеся не замечают ошибок в своих решениях. Рекомендуется приём «Проверка на правдоподобность», эффективный инструмент, который помогает учащимся оценивать реалистичность полученных ответов и выявлять ошибки в расчётах. Он учит связывать математические вычисления с реальным контекстом задачи, что особенно важно в информатике.

3. *Формирование навыков самоорганизации.* Учащиеся должны уметь организовать свою работу на экзамене: распределить время, определить, какие задания решать в первую очередь. В качестве рекомендаций больше выполнять тренировочные работы в формате КЕГЭ.

• **Формирование коммуникативных УУД (работа с информацией, формулирование мыслей).**

1. *Развитие умения работать с информацией, представленной в различных формах.* Учащиеся должны уметь извлекать информацию из текста, таблиц, схем и преобразовывать её из одной формы в другую. Рекомендуется учить учащихся читать и понимать инструкции, алгоритмы, технические описания, выделять в них ключевую информацию.

2. *Развитие умения формулировать и аргументировать свои мысли.* В заданиях с кратким ответом важно уметь кратко и точно сформулировать ответ.

Использовать приём «Объяснение решения». После выполнения задания требовать от учащихся устного объяснения своего решения, аргументации каждого шага. Приём «Запись решения с комментариями». Требовать от учащихся записи решения на черновике с комментариями, поясняющими каждое действие.

• **Рекомендации по организации учебного процесса.**

1. *Дифференциация и индивидуализация обучения.* Для учащихся данной группы необходима дифференциация заданий и индивидуальный подход. Разработать разноуровневые задания для разных групп учащихся. Для слабых учащихся – базовые задания с пошаговыми инструкциями, для сильных – более сложные и творческие.

2. *Систематичность и регулярность повторения.* Для закрепления знаний необходимо регулярное повторение ранее изученного материала. На каждом уроке выделять 5–10 минут на повторение ранее изученного материала (устный счёт, алгоритмические диктанты, короткие проверочные работы). Использовать систему опорных конспектов, которые позволяют быстро повторить основные понятия и формулы.

3. *Создание ситуации успеха и мотивация.* Важно формировать у учащихся уверенность в своих силах и положительную мотивацию к изучению предмета. Отмечать и поощрять даже малейшие успехи, хвалить за правильно решённые задания, за попытки решить сложную задачу. Использовать положительное подкрепление: за каждые 5 правильно решённых заданий давать небольшие поощрения. Создавать ситуацию успеха, давая в начале обучения более простые задания и постепенно усложняя их.

• **Рекомендации по использованию цифровых образовательных ресурсов.**

Активно использовать открытый банк заданий ФИПИ для систематической тренировки. Отобрать для каждой «зоны роста» по 20–30 типовых заданий и решать их в классе и дома. Использовать электронные образовательные платформы (например, «КЕГЭ», «Решу ЕГЭ», «Яндекс.Учебник») для организации самостоятельной работы учащихся и автоматической проверки результатов. Использовать среды визуального программирования (например, «КуМир», Python Turtle) для развития алгоритмического мышления и визуализации алгоритмов.

Основное направление работы с учащимися, рискующими не преодолеть минимальный порог, заключается в смещении акцента с изучения всего объёма материала на освоение ключевых, базовых тем, которые гарантируют получение минимальных баллов. При этом важнейшей задачей является не просто передача знаний, а формирование метапредметных умений: умения читать и понимать условие задачи, планировать свои действия, контролировать результат, анализировать информацию и строить логические рассуждения. Эти умения являются универсальными и будут полезны не только для сдачи ЕГЭ, но и для дальнейшего обучения и профессиональной деятельности. Работа должна быть систематической, регулярной и терпеливой, с обязательным учётом индивидуальных особенностей каждого учащегося.

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

- **Формирование познавательных УУД (работа с информацией, логические операции, моделирование).**

1. *Развитие логического мышления и работы с абстрактными моделями.*
Для преодоления дефицитов, связанных с заданиями на логику и системы счисления, необходимо систематически развивать логическое мышление. Рекомендуется применение законов алгебры логики в повседневной практике. На каждом уроке, где встречаются логические задачи, требовать от учащихся упрощения выражений с использованием законов и обоснования каждого шага преобразования. Использовать задания на нахождение ошибок в упрощении выражений, что развивает критическое мышление. Использовать метод «Логических цепочек» для сложных выражений. При решении задач на логику (№ 2, № 15) требовать от учащихся записи полной цепочки рассуждений с пояснением каждого шага и указанием применяемого закона. Например: «Используем закон де Моргана для $\neg (A \wedge B)$, получаем $(\neg A \vee \neg B)$ ». Решение комбинированных задач на логику и системы счисления.

2. *Развитие навыков алгоритмического мышления и программирования.*
Для преодоления дефицитов, связанных с заданиями на алгоритмизацию и программирование, необходимо углублять навыки алгоритмического мышления и практического программирования. Рекомендуется использовать метод «Пошагового исполнения алгоритма с трассировкой» для сложных алгоритмов. При разборе алгоритмов с вложенными циклами и рекурсией требовать от учащихся детальной трассировки с фиксацией значений всех переменных на каждом шаге. Использовать трассировочные таблицы с несколькими уровнями вложенности. Решение задач на программирование с пошаговым усложнением. Начинать с простых программ на 10–15 строк (задание № 17), постепенно усложняя их до программ на 20–30 строк (задание № 25, № 26). Использовать задачи на обработку массивов, строк, файлов. Организация работы с файлами в программах. Учить учащихся организовывать ввод данных из файлов и вывод результатов, что необходимо для выполнения заданий № 26 и № 27.

- **Формирование регулятивных УУД (планирование, самоконтроль, самооценка).**

1. *Развитие навыков планирования деятельности при решении комбинированных задач.* Учащиеся данной группы не умеют эффективно планировать свои действия при решении многошаговых задач, что приводит к потере времени и ошибкам. Рекомендуется использовать приём «Детальный план решения задачи». Перед решением каждой сложной задачи требовать от учащихся составления письменного плана с указанием всех шагов и промежуточных результатов. План должен включать не только последовательность действий, но и указание, какие формулы или алгоритмы

будут использованы на каждом шаге. Приём «Сравнение способов решения». Предлагать учащимся решить одну и ту же задачу разными способами и сравнить их эффективность. Это развивает умение выбирать оптимальный путь решения.

2. *Развитие навыков самоконтроля и самопроверки.* Для предотвращения потери баллов из-за невнимательности и ошибок в вычислениях необходимо развивать навыки самоконтроля. Рекомендуется приём «Многоуровневая проверка». Учитывать учащихся проверять свои решения на нескольких уровнях: 1) проверка вычислений, 2) проверка логики решения, 3) проверка на правдоподобность, 4) проверка соответствия формату ответа. Использование критериев оценивания заданий повышенного уровня. Знакомить учащихся с критериями оценивания заданий с развёрнутым ответом, чтобы они понимали, за что снимаются баллы, и могли проверить себя по этим критериям.

3. *Формирование навыков самоорганизации и управления временем.* Учащиеся должны уметь организовать свою работу на экзамене для успешного выполнения большего количества заданий. Рекомендуется проводить тренировочные работы с ограничением по времени, организовывать регулярные тренировочные работы, приближенные к реальным условиям экзамена, с жёстким контролем времени на выполнение каждого задания.

• **Формирование коммуникативных УУД (работа с информацией, формулирование мыслей).**

1. *Развитие смыслового чтения и работы со сложными текстами.* Учащиеся должны уметь понимать сложные условия задач, выделять ключевую информацию и интерпретировать её. Рекомендуется использовать приём «Структурирование условия задачи». Учитывать учащихся представлять условие задачи в виде схемы, таблицы или краткого конспекта, выделяя данные, условия и вопрос. Это помогает лучше понять задачу и не упустить важные детали. Внедрение приёма «Переформулирование задачи в общем виде». После прочтения условия просить учащихся сформулировать задачу в общем виде, используя математические или логические обозначения. Например, «Найти все числа N в диапазоне от 1 до 1000, для которых верно, что ...».

2. *Развитие умения формулировать и аргументировать свои мысли.* Учащиеся должны уметь кратко и точно формулировать ответы, особенно в заданиях с кратким ответом. Рекомендуется использование приёма «Объяснение решения с аргументацией». После выполнения задания повышенной сложности требовать от учащихся устного или письменного объяснения своего решения с аргументацией каждого шага. Это помогает структурировать мысли и выявлять ошибки в рассуждениях. Приём «Краткий и точный ответ». Тренировать умение формулировать краткие и точные ответы в соответствии с требованиями задания. Обсуждать, как можно сформулировать ответ более лаконично и точно.

• Рекомендации по организации учебного процесса.

1. *Дифференциация и индивидуализация обучения.* Для учащихся данной группы необходима дифференциация заданий и индивидуальный подход с акцентом на развитие навыков решения задач повышенной сложности. Разработать разноуровневые задания для разных групп учащихся. Для данной группы – задания повышенного уровня с постепенным усложнением. Проводить индивидуальные консультации для каждого учащегося, анализировать его ошибки и давать индивидуальные задания для их устранения.

2. *Систематичность и регулярность повторения.* Для закрепления знаний и развития навыков необходимо регулярное повторение и систематическая тренировка. На каждом уроке выделять 10–15 минут на решение задач повышенного уровня сложности с разбором каждого шага. Использовать систему опорных конспектов, которые позволяют быстро повторить основные понятия и формулы. Организовать систему накопительного оценивания, где за решение задач повышенного уровня учащиеся получают дополнительные баллы, что стимулирует их к систематической работе.

3. *Организация проектной деятельности.* Для развития навыков решения практических задач и алгоритмического мышления рекомендуется организация проектной деятельности. Организовать проектную деятельность по созданию программ для обработки данных (работа с файлами, массивами, строками). Рекомендации по использованию цифровых образовательных ресурсов.

Активно использовать открытый банк заданий ФИПИ для систематической тренировки, отбирая задания повышенного уровня сложности. Использовать электронные образовательные платформы (например, «КЕГЭ», «Решу ЕГЭ», «Яндекс.Учебник») для организации самостоятельной работы учащихся и автоматической проверки результатов. Использовать среды программирования (Python, PascalABC) для практической отработки навыков программирования и решения задач с файлами.

Основная стратегия работы с учащимися, показавшими результаты от минимального до 60 баллов, заключается в углублении знаний и развитии навыков применения их в нестандартных ситуациях. Цель – уверенное получение 60–70 тестовых баллов. При этом важнейшей задачей является не просто передача знаний, а формирование метапредметных умений: умения анализировать информацию, устанавливать причинно-следственные связи, строить логические рассуждения, планировать свои действия и контролировать результат. Работа должна быть систематической, регулярной и целенаправленной, с постепенным усложнением заданий и обязательным учётом индивидуальных особенностей каждого учащегося. Особое внимание следует уделить отработке навыков решения задач повышенного уровня сложности, что позволит учащимся перейти на следующий уровень подготовки.

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

- **Формирование познавательных УУД (работа с информацией, логические операции, моделирование).**

- *Развитие навыков алгоритмического мышления и составления программ.* Для преодоления дефицитов, связанных с заданиями на программирование высокого уровня сложности, необходимо углублять навыки алгоритмического мышления и практического программирования. Рекомендуется использовать метод «Проектирование программы сверху вниз» (нисходящее проектирование). Учитывать учащихся начинать решение задачи с разработки общей структуры программы, а затем детализировать каждый шаг. Использовать методы пошаговой детализации, псевдокод и блок-схемы для планирования программы. Внедрение приёма «Анализ эффективности алгоритмов». При разборе каждого алгоритма требовать от учащихся оценки его временной и пространственной сложности. Сравнивать различные алгоритмы для одной задачи и обсуждать, какой из них является оптимальным в зависимости от размера входных данных. Особое внимание уделить заданиям № 26 и № 27, где требуется обработка больших объёмов данных. Давать учащимся неэффективный код и просить их оптимизировать его, объясняя, почему новый вариант работает быстрее и эффективнее. Это развивает понимание принципов оптимизации алгоритмов и умение анализировать программный код.

- *Развитие навыков обработки символьной информации.* Учащиеся должны научиться эффективно обрабатывать символьную информацию, используя строковые методы и функции. Рекомендуется практикумы по обработке строк. Организовать серию практических работ по обработке строк с использованием различных методов и функций языка программирования. Начать с задач на поиск и замену подстрок, затем перейти к задачам на анализ последовательностей символов с использованием конечных автоматов.

Приём «Анализ текстовых данных». Использовать реальные текстовые данные (журналы, логи, текстовые файлы) для отработки навыков обработки символьной информации. Это помогает учащимся понять практическую значимость задачи и лучше усвоить методы её решения.

- *Развитие навыков моделирования и формализации.* Учащиеся должны научиться формализовать практические задачи и строить адекватные математические модели. Рекомендуется использовать метод «Формализация задачи». При решении каждой практической задачи требовать от учащихся определения ключевых параметров модели, переменных, ограничений и целевой функции. Записывать модель в математической форме.

- **Формирование регулятивных УУД (планирование, самоконтроль, самооценка).**

1. *Развитие навыков планирования сложной деятельности.* Учащиеся должны научиться планировать решение многоэтапных задач, особенно в

области программирования и анализа данных. Рекомендуется приём «Детальное проектирование решения». Перед написанием программы требовать от учащихся составления детального плана решения с указанием всех этапов, структур данных, алгоритмов и ожидаемых результатов. Использование метода «Пошаговая реализация с контролем». Требовать от учащихся пошаговой реализации программы с проверкой результата на каждом этапе. Это помогает выявлять ошибки на ранних стадиях и избегать их накопления.

2. *Развитие навыков самоконтроля и тестирования программ.* Учащиеся должны научиться систематически проверять свои программы на различных наборах данных. Рекомендуется метод «Разработка тестов». Учить учащихся разрабатывать наборы тестовых данных для проверки своих программ, включая стандартные, критические и граничные случаи. Использовать требования к формату входных и выходных данных. Приём «Тестирование на граничных значениях». Особое внимание уделять проверке программ на граничных значениях входных данных (минимальные и максимальные значения, пустые файлы, нулевые значения и т.д.).

3. *Развитие навыков работы с большими объёмами данных.* Учащиеся должны научиться обрабатывать большие объёмы данных эффективно и без потерь. Рекомендуется метод «Оценка объёма данных». Учить учащихся оценивать объём входных данных и выбирать алгоритмы, которые могут эффективно обрабатывать данные такого размера. Приём «Оптимизация ввода-вывода». Учить учащихся организовывать эффективный ввод-вывод данных из файлов, особенно для больших объёмов (десятки и сотни тысяч строк).

• **Формирование коммуникативных УУД (работа с информацией, формулирование мыслей).**

1. *Развитие навыков смыслового чтения сложных текстов.* Учащиеся должны уметь понимать сложные условия задач, выделять ключевую информацию и интерпретировать её. Рекомендуется метод «Структурный анализ условия». Учить учащихся представлять сложное условие задачи в виде схемы, таблицы или краткого конспекта, выделяя входные данные, требования, ограничения и ожидаемый результат. Наиболее эффективен будет приём «Перевод условия на математический язык». Требовать от учащихся перевода сложных условий задач на язык математических и логических формул.

2. *Развитие умения документировать программный код.* Учащиеся должны уметь писать структурированный, документированный код, который легко читать и проверять. Рекомендуется использовать приём «Комментирование программы». Требовать от учащихся написания комментариев к каждой логической части программы, объясняющих, что делает эта часть кода. Приём «Структурирование программы». Учить учащихся использовать функции и процедуры для структурирования программы, что делает её более читаемой и понятной для проверяющего.

• **Рекомендации по организации учебного процесса.**

1. *Углублённое изучение алгоритмов и структур данных.* Для учащихся данной группы необходимо углублённое изучение алгоритмов и структур данных, выходящее за рамки школьной программы. Изучать различные

алгоритмы сортировки, поиска, обработки строк и массивов, анализировать их эффективность. Изучать методы динамического программирования, жадные алгоритмы, рекурсивные и итеративные алгоритмы. Изучать структуры данных: списки, стеки, очереди, деревья и их применение для решения практических задач. олимпиадные задачи для развития навыков решения сложных алгоритмических задач.

2. *Организация практикумов по программированию.* Для развития навыков практического программирования необходимо организовать регулярные практикумы. Проводить еженедельные практические занятия по программированию на 2–3 часа, где учащиеся решают задачи высокого уровня сложности (№ 24–27). Использовать автоматическую проверку решений на различных наборах данных. Организовывать соревнования и конкурсы по программированию для повышения мотивации.

3. *Подготовка к олимпиадам и участие в конкурсах.* Для развития нестандартного мышления и углублённых знаний алгоритмов рекомендуется участие в олимпиадах и конкурсах. Организовывать подготовку к олимпиадам по информатике различного уровня. Использовать олимпиадные задачи на уроках и факультативных занятиях. Участвовать в онлайн-олимпиадах и конкурсах по программированию.

• **Рекомендации по использованию цифровых образовательных ресурсов.**

Активно использовать открытый банк заданий ФИПИ для систематической тренировки, отбирая задания повышенного и высокого уровня сложности. Использовать электронные образовательные платформы (например, «Онлайн школа Фокс форд», «Решу ЕГЭ», «Яндекс. Учебник») для организации самостоятельной работы учащихся с автоматической проверкой результатов. Использовать онлайн-среды разработки (IDE) и системы контроля версий для организации совместной работы над проектами. Создать для учащихся библиотеку эффективных алгоритмов и шаблонов решений типовых задач высокого уровня сложности.

Основная стратегия работы с учащимися, показавшими результаты от 61 до 80 баллов, заключается в углублении знаний в области алгоритмизации и программирования, а также развитии навыков решения комплексных задач высокого уровня сложности. Цель – стабильное получение 80+ тестовых баллов. При этом важнейшей задачей является не просто передача знаний, а формирование метапредметных умений: умения моделировать и прогнозировать, оценивать эффективность алгоритмов, планировать свою деятельность и осуществлять самоконтроль. Работа должна быть систематической, регулярной и целенаправленной, с акцентом на практическое программирование и решение задач, требующих глубокого алгоритмического мышления. Особое внимание следует уделить отработке навыков обработки символьной и целочисленной информации, а также методам анализа данных, что позволит учащимся перейти в группу высокобалльников и достичь 80 и более баллов на ЕГЭ.

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

- **Формирование познавательных УУД (работа с информацией, логические операции, моделирование).**

1. *Развитие навыков моделирования сложных процессов.* Для преодоления дефицитов, связанных с заданиями на программирование высокого уровня сложности, необходимо развивать навыки моделирования и абстрагирования. Рекомендуется использовать метод «Построение абстрактных моделей». Учитывать учащихся строить абстрактные модели для решения сложных задач, особенно в области обработки символьной информации. Например, для задания № 24 учить строить конечные автоматы для поиска сложных паттернов в строках. Разбирать различные примеры построения конечных автоматов для разных типов задач. Подключение приёма «Прогнозирование результатов». При решении задач на анализ данных (задание № 27) учить учащихся прогнозировать возможные результаты и проверять свои прогнозы на различных наборах данных.

2. *Развитие навыков оценки эффективности алгоритмов.* Учащиеся должны научиться оценивать эффективность алгоритмов и выбирать оптимальные решения для различных задач. Рекомендуется практическая работа на основе приема «Анализ сложности алгоритмов». При разборе каждого алгоритма требовать от учащихся оценки его временной и пространственной сложности. Сравнить различные алгоритмы для одной задачи и обсуждать, какой из них является оптимальным в зависимости от размера входных данных. Приём «Оптимизация алгоритмов». Давать учащимся неэффективный код и просить их оптимизировать его, объясняя, почему новый вариант работает быстрее и эффективнее.

3. *Развитие навыков обработки символьной информации.* Учащиеся должны научиться эффективно обрабатывать символьную информацию, используя различные методы и подходы. Рекомендуется увеличить количество часов на практикум по обработке строк. Организовать серию практических работ по обработке строк с использованием различных методов и функций языка программирования. Включить задачи на построение конечных автоматов, поиск подстрок, анализ последовательностей символов. Включение метода «Конечные автоматы в программировании». Учитывать учащихся строить конечные автоматы для решения задач на обработку символьной информации. Использовать визуализацию состояний для лучшего понимания работы алгоритма. Выполнить подборку «Решение нестандартных задач». Использовать задачи из олимпиад и конкурсов по программированию для развития навыков обработки символьной информации в нестандартных ситуациях.

• **Формирование регулятивных УУД (планирование, самоконтроль, самооценка).**

1. *Развитие навыков управления временем и самоорганизации.* Учащиеся должны научиться эффективно распределять время на экзамене, чтобы успеть выполнить все задания. Рекомендуется выполнять тренировочные работы с ограничением по времени. Проводить регулярные тренировочные работы в условиях, максимально приближенных к реальному экзамену, с жёстким контролем времени на выполнение каждого задания. Анализировать, сколько времени тратится на каждое задание, и разрабатывать стратегии оптимизации времени. Выстроить «Стратегия выполнения работы». Разработать с учащимися индивидуальную стратегию выполнения работы, учитывающую их сильные и слабые стороны. Обсудить, на какие задания тратить больше времени, а от каких можно отказаться в пользу других.

2. *Развитие навыков самоконтроля и тестирования программ.* Учащиеся должны научиться систематически проверять свои программы на различных наборах данных. Рекомендуется воспользоваться приёмом «Разработка тестов». Учить учащихся разрабатывать наборы тестовых данных для проверки своих программ, включая стандартные, критические и граничные случаи. Особое внимание уделять проверке на больших объёмах данных, чтобы убедиться в эффективности алгоритма. Активно внедрять «Тестирование на скрытых данных». Учить учащихся предвидеть возможные скрытые тесты и проверять свои программы на таких данных. Использовать автоматическую проверку решений на различных наборах данных. Работать по приему «Самопроверка по критериям». Знакомить учащихся с критериями оценивания заданий с развёрнутым ответом и требовать проверки своих решений по этим критериям перед сдачей работы.

3. *Развитие навыков работы с большими объёмами данных и файлами.* Учащиеся должны научиться обрабатывать большие объёмы данных эффективно и без потерь. Рекомендуется использовать приём «Оценка объёма данных». Учить учащихся оценивать объём входных данных и выбирать алгоритмы, которые могут эффективно обрабатывать данные такого размера. Особое внимание уделять оценке эффективности алгоритмов по времени и памяти. «Оптимизация ввода-вывода». Учить учащихся организовывать эффективный ввод-вывод данных из файлов, особенно для больших объёмов (десятки и сотни тысяч строк).

• **Формирование коммуникативных УУД (работа с информацией, формулирование мыслей).**

1. *Развитие навыков смыслового чтения сложных текстов.* Учащиеся должны уметь понимать сложные условия задач, выделять ключевую информацию и интерпретировать её. Производить детальный анализ условия, учить учащихся детально анализировать условие задачи, выделять все требования, ограничения и ожидаемые результаты. Использовать метод постановки вопросов к условию для выявления всех нюансов. Требовать от учащихся перевода сложных условий задач на язык алгоритмов и программного кода.

2. *Развитие умения структурировать программный код.* Учащиеся должны уметь писать структурированный, документированный код, который легко читать и проверять. Знакомить учащихся со стандартами оформления кода, требовать их соблюдения при написании программ для заданий с развёрнутым ответом. Требовать от учащихся написания комментариев к сложным алгоритмам, объясняющих логику их работы. Это помогает не только проверяющему, но и самому учащемуся лучше понять свой алгоритм.

• Рекомендации по организации учебного процесса.

1. *Углублённое изучение алгоритмов и структур данных за рамками школьной программы.* Для учащихся данной группы необходимо углублённое изучение алгоритмов и структур данных, выходящее за рамки школьной программы.

- Изучать сложные алгоритмы: алгоритмы на графах (поиск в глубину, поиск в ширину), динамическое программирование, алгоритмы на строках.

- Изучать структуры данных: деревья, стеки, очереди, списки, их применение для решения практических задач.

- Использовать олимпиадные задачи для развития навыков решения сложных алгоритмических задач.

- Организовать факультативные занятия по углублённому изучению алгоритмов и структур данных.

2. *Организация олимпиадной подготовки.* Для развития навыков решения нестандартных задач рекомендуется организация олимпиадной подготовки.

- Организовывать регулярные тренировки по решению олимпиадных задач.

- Участвовать в олимпиадах по информатике различного уровня (школьных, муниципальных, региональных, всероссийских).

- Использовать задачи с онлайн-платформ (Яндекс. Учебник, КЕГЭ, Решу ЕГЭ) для тренировки.

3. *Проведение тренировочных работ в формате ЕГЭ.* Для развития навыков управления временем и адаптации к формату экзамена необходимо проводить регулярные тренировочные работы.

- Проводить тренировочные работы в условиях, максимально приближенных к реальному экзамену (ограничение по времени, запрет на использование интернета, работа в специализированном программном обеспечении).

- Анализировать результаты тренировочных работ и выявлять типичные ошибки.

- Разрабатывать стратегии выполнения работы с учётом индивидуальных особенностей каждого учащегося.

• Рекомендации по использованию цифровых образовательных ресурсов.

Активно использовать открытый банк заданий ФИПИ для систематической тренировки, отбирая задания высокого уровня сложности. Использовать электронные образовательные платформы (например, «КЕГЭ», «Решу ЕГЭ», «Яндекс Учебник») для организации самостоятельной работы

учащихся с автоматической проверкой результатов. Использовать онлайн-среды разработки (IDE) и системы контроля версий для организации совместной работы над проектами, платформы для решения олимпиадных задач для развития навыков решения сложных алгоритмических задач. Создать для учащихся библиотеку эффективных алгоритмов и шаблонов решений типовых задач высокого уровня сложности.

Основная стратегия работы с учащимися, показавшими результаты от 81 до 100 баллов, заключается в устранении точечных дефицитов и доведении навыков решения задач всех уровней сложности до совершенства. Главная цель – получение 100 тестовых баллов. При этом важнейшими задачами являются: углублённое изучение алгоритмов и структур данных, выходящее за рамки школьной программы; развитие навыков построения сложных абстрактных моделей для решения задач на обработку символьной информации (конечные автоматы); совершенствование навыков оценки эффективности алгоритмов и выбора оптимальных решений для больших объёмов данных; отработка внимательности и аккуратности при выполнении стандартных заданий; развитие навыков управления временем и самоорганизации; психологическая подготовка к экзамену. Работа должна быть систематической, регулярной и целенаправленной, с акцентом на индивидуальные особенности каждого учащегося и устранение его личных «слабых мест». Особое внимание следует уделить отработке навыков решения заданий высокого уровня сложности (№ 24–27), что позволит учащимся достичь 100 баллов на ЕГЭ.

**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**

Рекомендуем проведение практических семинаров для учителей информатики с привлечением старших и ведущих экспертов региональной предметной комиссии ЕГЭ по информатике, а также председателя региональной предметной комиссии ЕГЭ по информатике. На таких семинарах необходимо обсудить подходы к проверке всех заданий с развернутым ответом ЕГЭ по информатике.

Организовать обмен опытом по подготовке выпускников к решению задач с развернутым ответом, в том числе задач по программированию повышенного уровня сложности, и использование Банка задач ФИПИ по информатике для подготовки к выполнению выпускниками заданий с кратким ответом. Особое внимание на практических семинарах обратить на преемственность курса информатики в школе, а также на выполнение требований ФГОС ОО и ФГОС СОО.

Учителям, собирающимся работать в 10–11 классах в 2026–2027 учебном году, необходимо провести поэлементный анализ заданий, традиционно вызывающих затруднения у выпускников, используя аналитические отчеты региона и методические рекомендации прошлых лет.

Организация методического сопровождения учителей информатики с привлечением ведущих специалистов, преподавателей кафедры математики и информатики ГАУ ДПО РО «Институт развития образования». Рекомендуется организовать обсуждение следующих актуальных тем на методических объединениях учителей информатики:

1. «Анализ результатов ЕГЭ по информатике 2026 и выявление типичных ошибок и затруднений». Обсуждение основных тенденций, выявленных дефицитов и зон роста.

2. «Формирование метапредметных умений на уроках информатики как основа успешной сдачи ЕГЭ». Рассмотрение конкретных методик и приёмов работы с текстом, развития логического и алгоритмического мышления.

3. «Эффективные практики подготовки к решению заданий с развёрнутым ответом (№ 24–27)». Трансляция опыта учителей, чьи ученики успешно справляются с заданиями высокого уровня сложности, обсуждение методических подходов к обучению программированию.

4. «Методика работы с учащимися разных групп: от минимального порога до 100 баллов». Обсуждение подходов к дифференцированному обучению, разработки индивидуальных образовательных маршрутов и системы мониторинга достижений.

5. «Использование современных педагогических технологий для повышения мотивации к изучению информатики». Обмен опытом по вовлечению учащихся в проектную, исследовательскую и олимпиадную деятельность, использованию цифровых образовательных ресурсов.