ГЛАВА 2.
Методический анализ результатов ЕГЭ

по ИНФОРМАТИКЕ

РАЗДЕЛ 1. ХАРАКТЕРИСТИКА УЧАСТНИКОВ ЕГЭ ПО УЧЕБНОМУ ПРЕДМЕТУ

1.1.Количество участников ЕГЭ по учебному предмету (за 3 года)

Таблица 2-1

2024 г.		2025 г.		2026 г.	
чел.	% от общего числа участников	чел.	% от общего числа участников	чел.	% от общего числа участников
591	17,08	742	20,12	760	20,86

1.2.Процентное соотношение юношей и девушек, участвующих в ЕГЭ (за 3 года)

Таблица 2-2

Пол	2024 г.		2025 г.		2026 г.	
	чел.	% от общего числа участников	чел.	% от общего числа участников	чел.	% от общего числа участников
Женский	179	30,29	215	28,98	216	28,42
Мужской	412	69,71	527	71,02	544	71,58

1.3. Количество участников экзамена в регионе по категориям (за 3 года)

Таблица 2-3

Категория участика	2024 г.		2025 г.		2026 г.	
	чел.	% от общего числа участников	чел.	% от общего числа участников	чел.	% от общего числа участников
ВТГ, обучающиеся по программам СОО	586	99,15	738	99,46	758	99,74

ВПЛ	1	0,17	0	0,00	0	0,00
ВТГ, обучающиеся по программам СПО	4	0,69	4	0,54	2	0,26
Обучающийся общеобразовательной организации, завершивший освоение образовательной программы по учебному предмету	0	0,00	0,00	0,00	0	0,00

1.4.Количество участников экзамена в регионе по типам ОО

Таблица 2-4

№ п/п	Категория участика	2024 г.		2025 г.		2026 г.	
		чел.	% от общего числа участников	чел.	% от общего числа участников	чел.	% от общего числа участников
1.	СОШ	273	46,51	378	51,22	410	54,09
2.	СОШ с УИОП	126	21,47	149	20,19	130	17,15
3.	Гимназии	51	8,69	73	9,89	71	9,37
4.	Лицеи	136	23,17	136	18,43	144	19,00
6.	Кадетские школы	0	0,00	2	0,27	3	0,40

1.5.Количество участников ЕГЭ по учебному предмету по АТЕ региона

Таблица 2-5

№ п/п	Наименование АТЕ	Количество участников ЕГЭ по учебному предмету	% от общего числа участников в регионе
1.	МО ГО «Сыктывкар»	267	35,13
2.	МО «Воркута» Республики Коми	44	5,79
3.	МО «Вуктыл» Республики Коми	13	1,71
4.	МО «Инта»	11	1,45
5.	МР «Печора»	22	2,89
6.	МР «Сосногорск»	32	4,21
7.	МО «Усинск» Республики Коми	52	6,84
8.	МО «Ухта» Республики Коми	166	21,84

9.	МР «Ижемский»	18	2,37
10.	МР «Княжпогостский»	11	1,45
11.	МР «Койгородский»	1	0,13
12.	МР «Корткеросский»	4	0,53
13.	МР «Прилузский»	17	2,24
14.	МР «Сыктывдинский» Республики Коми	10	1,32
15.	МР «Сысольский»	4	0,53
16.	МР «Троицко-Печорский»	6	0,79
17.	МР «Удорский»	4	0,53
18.	МР «Усть-Вымский»	14	1,84
19.	МР «Усть-Куломский»	12	1,58
20.	МР «Усть-Цилемский»	11	1,45
21.	ГОУ «Гимназия искусств при Главе Республики Коми»	1	0,13
22.	ГОУ «КРЛ при СГУ»	16	2,11
23.	ГОУ РК «ФМЛИ»	13	1,71
24.	ГОУ «Лицей для одаренных детей»	11	1,45

1.6. Прочие характеристики участников экзаменационной кампании (при наличии)

Дополнительных характеристик участников экзаменационной кампании не выявлено. Все участники сдавали экзамен в штатном режиме без особенностей.

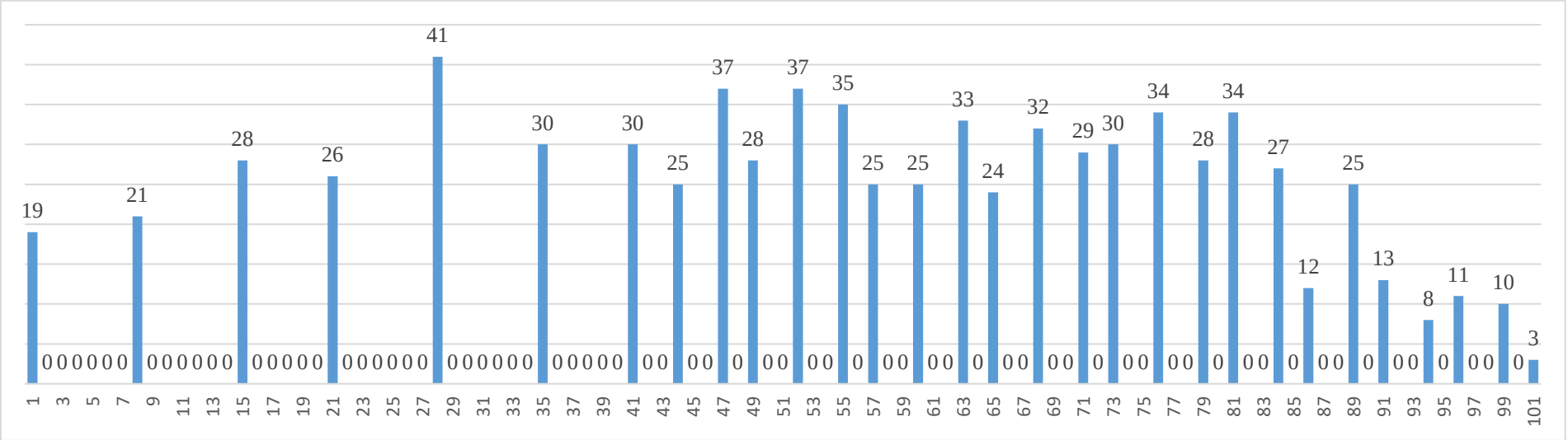
1.7.ВЫВОДЫ о характере изменения количества участников ЕГЭ по учебному предмету

В 2026 году в ЕГЭ по информатике приняло участие 760 человек, что на 18 человек больше, чем в 2025 году (742) и на 169 человек больше, чем в 2024 году (591). Наблюдается устойчивая положительная динамика количества участников экзамена (рост на 28,6% за три года). Данная тенденция связана с повышением интереса выпускников к IT-специальностям, а также с расширением перечня направлений подготовки в вузах, требующих результатов ЕГЭ по информатике. Соотношение юношей и девушек сохраняется на уровне примерно 72% к 28% соответственно. Основную массу участников составляют выпускники текущего года (99,74%). Среди типов ОО преобладают обучающиеся общеобразовательных школ (54,09%), лицеев (19,00%) и гимназий (17,15%).

РАЗДЕЛ 2. ОСНОВНЫЕ РЕЗУЛЬТАТЫ ЕГЭ ПО ПРЕДМЕТУ

2.1. Диаграмма распределения тестовых баллов участников ЕГЭ по предмету в 2026 г.

(количество участников, получивших тот или иной тестовый балл)



2.2.Динамика результатов ЕГЭ по предмету за последние 3 года

Таблица 2-6

№ п/п	Участников, набравших балл	Год проведения ГИА		
		2024 г.	2025 г.	2026 г.
1.	ниже минимального балла, %	17,06	22,91	21,71
2.	от минимального балла до 60 баллов, %	35,14	36,93	31,84
3.	от 61 до 80 баллов, %	35,98	28,57	32,11
4.	от 81 до 100 баллов, %	11,82	11,59	14,34
5.	Средний тестовый балл	56,69	53,71	55,56

2.3.Результаты ЕГЭ по учебному предмету по группам участников экзамена с различным уровнем подготовки

2.3.1. в разрезе категорий участников ЕГЭ

Таблица 2-7

№ п/п	Категории участников	Доля участников, у которых полученный тестовый балл			
		ниже минимального	от минимального балла до 60 баллов	от 61 до 80 баллов	от 81 до 100 баллов
1.	ВТГ, обучающиеся по программам СОО	21,64	31,79	32,19	14,38
2.	ВТГ, обучающиеся по программам СПО	50,00	50,00	0,00	0,00
3.	ВПЛ	0,00	0,00	0,00	0,00
4.	Обучающийся общеобразовательной организации, завершивший освоение образовательной программы по учебному предмету	0,00	0,00	0,00	0,00

2.3.2. в разрезе типа ОО

Таблица 2-8

№ п/п	Тип ОО	Количество участников, чел.	Доля участников, получивших тестовый балл			
			ниже минимального	от минимального до 60 баллов	от 61 до 80 баллов	от 81 до 100 баллов
1.	СОШ	410	28,54	36,83	28,29	6,34
2.	СОШ с УИОП	130	24,62	29,23	36,15	10,00
3.	гимназии	71	12,68	33,80	40,85	12,68
4.	лицеи	144	2,78	18,75	36,11	42,36
5.	кадетские школы	3	66,67	33,33	0,00	0,00

2.3.3. в разрезе уровня изучения предмета

Таблица 2-9

№ п/п	Уровень изучения предмета	Количество участников, чел.	Доля участников, получивших тестовый балл			
			ниже минимального	от минимального до 60 баллов	от 61 до 80 баллов	от 81 до 100 баллов
1.	базовый	758	21,64	31,79	32,19	14,38
2.	углубленный	0	0,00	0,00	0,00	0,00

2.3.4. в сравнении по АТЕ

Таблица 2-10

№ п/п	Наименование АТЕ	Количество участников, чел.	Доля участников, получивших тестовый балл			
			ниже минимального	от минимального до 60 баллов	от 61 до 80 баллов	от 81 до 100 баллов
1.	МО ГО «Сыктывкар»	267	17,98	32,58	37,83	11,61
2.	МО «Воркута» Республики Коми	44	11,36	25,00	54,55	9,09
3.	МО «Вуктыл» Республики Коми	13	46,15	46,15	7,69	0,00
4.	МО «Инта»	11	27,27	45,45	18,18	9,09
5.	МР «Печора»	22	31,82	36,36	13,64	18,18
6.	МР «Сосногорск»	32	53,13	18,75	25,00	3,13
7.	МО «Усинск» Республики Коми	52	28,85	25,00	32,69	13,46
8.	МО «Ухта» Республики Коми	166	19,88	29,52	28,31	22,29
9.	МР «Ижемский»	18	16,67	50,00	22,22	11,11
10.	МР «Княжпогостский»	11	36,36	54,55	9,09	0,00
11.	МР «Койгородский»	1	0,00	100,00	0,00	0,00
12.	МР «Корткеросский»	4	25,00	0,00	75,00	0,00
13.	МР «Прилузский»	17	29,41	52,94	17,65	0,00

№ п/п	Наименование АТЕ	Количество участников, чел.	Доля участников, получивших тестовый балл			
			ниже минимального	от минимального до 60 баллов	от 61 до 80 баллов	от 81 до 100 баллов
14.	МР «Сыктывдинский» Республики Коми	10	30,00	10,00	50,00	10,00
15.	МР «Сысольский»	4	50,00	25,00	0,00	25,00
16.	МР «Троицко-Печорский»	6	0,00	50,00	50,00	0,00
17.	МР «Удорский»	4	50,00	50,00	0,00	0,00
18.	МР «Усть-Вымский»	14	21,43	64,29	7,14	7,14
19.	МР «Усть-Куломский»	12	8,33	16,67	41,67	33,33
20.	МР «Усть-Цилемский»	11	45,45	36,36	9,09	9,09
21.	ГОУ «Гимназия искусств при Главе Республики Коми»	1	0,00	100,00	0,00	0,00
22.	ГОУ «КРЛ при СГУ»	16	0,00	12,50	50,00	37,50
23.	ГОУ РК «ФМЛИ»	13	7,69	30,77	23,08	38,46
24.	ГОУ «Лицей для одаренных детей»	11	9,09	27,27	36,36	27,27

2.4.Выделение перечня ОО, продемонстрировавших наиболее высокие и низкие результаты ЕГЭ по предмету

2.4.1. Перечень ОО, продемонстрировавших наиболее высокие результаты ЕГЭ по предмету

Таблица 2-11

№ п/п	Наименование ОО	Количество ВТГ, чел.	Доля ВТГ, получивших тестовый балл			
			от 81 до 100 баллов	от 61 до 80 баллов	от минимального балла до 60 баллов	ниже минимального
1.	МАОУ «УТЛ» г. Ухта	31	80,65	16,13	3,23	0,00

№ п/п	Наименование ОО	Количество ВТГ, чел.	Доля ВТГ, получивших тестовый балл			
			от 81 до 100 баллов	от 61 до 80 баллов	от минимального балла до 60 баллов	ниже минимального
2.	МАОУ «Технологический лицей» г. Сыктывкар	13	61,54	38,46	0,00	0,00
3.	МАОУ «Лицей народной дипломатии» г. Сыктывкара	11	45,45	27,27	27,27	0,00

2.4.2. Перечень ОО, продемонстрировавших низкие результаты ЕГЭ по предмету

Таблица 2-12

№ п/п	Наименование ОО	Количество ВТГ, чел.	Доля ВТГ, получивших тестовый балл			
			ниже минимального	от минимального балла до 60 баллов	от 61 до 80 баллов	от 81 до 100 баллов
1.	МБОУ «СОШ № 2» г. Усинска	10	60,00	40,00	0,00	0,00

2.5. ВЫВОДЫ о характере изменения результатов ЕГЭ по предмету

В 2026 году средний тестовый балл по информатике составил 55,56, что выше показателя 2025 года (53,71) и ниже показателя 2024 года (56,69). Доля участников, не преодолевших минимальный балл, составила 31,84%, что на 5,09% меньше, чем в 2025 году (36,93%), что свидетельствует о повышении уровня базовой подготовки выпускников. Доля высокобалльников (81-100 т.б.) выросла до 14,34% (в 2025 г. – 11,59%, в 2024 г. – 11,82%), что является положительной динамикой. Доля участников, получивших от 61 до 80 баллов, составила 32,11% (в 2025 г. – 28,57%). Таким образом, наблюдается смещение распределения результатов в сторону более высоких баллов по сравнению с 2025 годом.

2.6. Связь динамики результатов ЕГЭ по предмету с принятыми на уровне субъекта Российской Федерации, муниципалитета, образовательных организаций мерами повышения качества освоения ФОП

Анализ результатов выполнения заданий позволяет сделать вывод о том, что положительная динамика (сохранение высоких показателей) результатов по отдельным заданиям достигнута благодаря в том числе рекомендациям, включенным в статистико-

аналитический отчет результатов ЕГЭ по учебному предмету в 2024-2025 году. При этом сохраняются темы, которые вызывают трудности, хотя в статистико-аналитический отчет по результатам ЕГЭ по учебному предмету в 2025 году были включены рекомендации и по этим темам (например, программирование).

Анализ профессиональной активности педагогов каждого муниципального образования, ежемесячно осуществляемый ГОУДПО «Коми республиканский институт развития образования» на основании статистики участия в методических мероприятиях, заседаниях республиканских методических объединений показывает, что активность педагогов недостаточно высокая. Педагоги не используют все возможности, которые реализованы в содержании мероприятий дорожной карты: участие в мероприятиях в режиме онлайн-подключения для удаленных территорий, возможность просмотра видеозаписи оффлайн в удобное время в сервисе «Видеозал», возможность задать вопросы и получить консультацию от специалистов, региональных и муниципальных методистов и др. При этом наблюдается взаимосвязь между низкой профессиональной активностью педагогов и низкими результатами обучающихся. В то же время отмечается повышение результатов в отдельных образовательных организациях и муниципальных образованиях, педагоги которых демонстрируют высокую профессиональную активность.

Можно отметить, что в дорожную карту были включены мероприятия, направленные на работу с образовательными организациями, учащиеся которых показали низкие результаты на экзамене. Данные мероприятия были реализованы. Образовательные организации, которые имели низкие показатели в прошлом году, в этом учебном году имеют более высокие результаты.

Кроме того, увеличению доли высокобалльников способствовала целенаправленная работа с мотивированными обучающимися в рамках профильного обучения и дополнительного образования.

Раздел 3. МЕТОДИЧЕСКИЙ АНАЛИЗ РЕЗУЛЬТАТОВ ВЫПОЛНЕНИЯ ЗАДАНИЙ КИМ[1] И РЕКОМЕНДАЦИИ ДЛЯ РЕГИОНАЛЬНОЙ СИСТЕМЫ ОБРАЗОВАНИЯ

3.1. Статистический анализ выполнения заданий КИМ

3.1.1. Основные статистические характеристики выполнения заданий КИМ в 2026 году

Основные статистические характеристики выполнения заданий в целом представлены в Таб.2-13. Информация о результатах оценивания выполнения заданий, в том числе в разрезе данных о получении того или иного балла по критерию оценивания выполнения каждого задания КИМ представлена в Таб. 2-14.

Таблица 2-13

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Республике Коми в группах участников экзамена с разными уровнями подготовки				
			средний, %	в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
1.	Умение представлять и считывать данные в разных типах информационных моделей (схемы, карты, таблицы, графики и формулы)	Б	88,08	62,35	93,55	95,14	99,13
2.	Умение строить таблицы истинности и логические схемы	Б	73,46	27,65	74,19	93,52	96,52
3.	Умение поиска информации в реляционных базах данных	Б	63,59	28,24	57,66	80,16	93,04
4.	Умение кодировать и декодировать информацию	Б	67,56	32,94	69,76	82,19	82,61

[1] При формировании отчетов по иностранным языкам рекомендуется выделять отдельные подразделы по устной и по письменной частям экзамена.

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Республике Коми в группах участников экзамена с разными уровнями подготовки				
			средний, %	в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
5.	Формальное исполнение простого алгоритма, записанного на естественном языке, или умение создавать линейный алгоритм для формального исполнителя с ограниченным набором команд, или умение восстанавливать исходные данные линейного алгоритма по результатам его работы	Б	56,41	4,71	41,94	86,64	99,13
6.	Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов	Б	60,51	14,71	54,44	83,40	92,17
7.	Умение определять объём памяти, необходимый для хранения графической и звуковой информации	Б	69,49	19,41	72,18	87,45	99,13
8.	Знание основных понятий и методов, используемых при измерении количества информации	Б	49,74	1,76	35,48	75,71	95,65
9.	Умение обрабатывать числовую информацию в электронных таблицах	Б	41,15	3,53	20,56	63,97	92,17
10.	Информационный поиск средствами текстового процессора	Б	78,72	50,59	79,84	88,26	97,39
11.	Умение подсчитывать информационный объём сообщения	П	40,38	2,94	25,00	59,92	86,96

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Республике Коми в группах участников экзамена с разными уровнями подготовки				
			средний, %	в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
12.	Умение исполнить алгоритм для конкретного исполнителя с фиксированным набором команд	П	53,59	4,12	43,55	78,54	94,78
13.	Умение использовать маску подсети	П	50,00	4,71	32,66	77,33	95,65
14.	Знание позиционных систем счисления	П	35,26	1,76	11,69	57,09	88,70
15.	Знание основных понятий и законов математической логики	П	35,13	2,35	14,92	53,04	88,70
16.	Вычисление рекуррентных выражений	П	60,13	4,71	51,61	88,66	99,13
17.	Умение составить алгоритм обработки числовой последовательности и записать его в виде простой программы (10–15 строк) на языке программирования	П	32,31	0,00	6,45	51,01	95,65
18.	Умение использовать электронные таблицы для обработки целочисленных данных	П	51,92	5,29	38,71	78,95	91,30
19.	Умение анализировать алгоритм логической игры	Б	54,74	7,65	39,52	82,19	98,26
20.	Умение найти выигрышную стратегию игры	П	50,13	0,59	26,21	85,83	98,26
21.	Умение построить дерево игры по заданному алгоритму и найти выигрышную стратегию	В	50,13	1,18	27,02	85,02	97,39

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Процент выполнения задания в Республике Коми в группах участников экзамена с разными уровнями подготовки				
			средний, %	в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
22.	Построение математических моделей для решения практических задач. Архитектура современных компьютеров. Многопроцессорные системы	П	38,97	3,53	20,97	57,89	89,57
23.	Умение анализировать ход исполнения алгоритма	П	30,38	0,59	4,44	48,58	91,30
24.	Умение создавать собственные программы (10–20 строк) для обработки символьной информации	В	6,28	0,00	0,00	0,81	40,87
25.	Умение создавать собственные программы (10–20 строк) для обработки целочисленной информации	В	11,03	0,00	0,00	7,29	59,13
26.	Умение обрабатывать целочисленную информацию с использованием сортировки	В	5,58	0,00	0,00	1,62	34,35
27.	Умение выполнять последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и построение модели, преобразование данных, визуализация данных, интерпретация результатов	В	9,74	0,00	0,00	3,44	58,70
Всего заданий – **27**; из них по уровню сложности: Б – **11**, П – **11**, В – **5**. Максимальный первичный балл за работу – **29**. Общее время выполнения работы – **3 часа 55 минут (235 мин.)**.							

Таблица 2-14

Номер задания / критерия оценивания в КИМ	Количество полученных первичных баллов	Процент участников экзамена в субъекте Российской Федерации, получивших соответствующий первичный балл за выполнения задания в группах участников экзамена с разными уровнями подготовки			
		в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б., %	в группе от 61 до 80 т.б., %	в группе от 81 до 100 т.б., %
1-1	0	38,18	6,61	4,92	0,92
1-1	1	61,82	93,39	95,08	99,08
1-2	0	72,73	25,62	6,56	3,67
1-2	1	27,27	74,38	93,44	96,33
1-3	0	72,73	42,98	19,67	7,34
1-3	1	27,27	57,02	80,33	92,66
1-4	0	68,48	30,17	18,03	18,35
1-4	1	31,52	69,83	81,97	81,65
1-5	0	95,15	57,85	13,52	0,92
1-5	1	4,85	42,15	86,48	99,08
1-6	0	84,85	45,45	15,98	5,5
1-6	1	15,15	54,55	84,02	94,5
1-7	0	80,61	28,1	12,7	0,92
1-7	1	19,39	71,9	87,3	99,08
1-8	0	98,18	63,64	24,59	4,59
1-8	1	1,82	36,36	75,41	95,41
1-9	0	96,36	79,75	35,66	7,34
1-9	1	3,64	20,25	64,34	92,66
1-10	0	48,48	18,6	11,89	2,75
1-10	1	51,52	81,4	88,11	97,25
1-11	0	96,97	74,38	40,16	13,76
1-11	1	3,03	25,62	59,84	86,24

Номер задания / критерия оценивания в КИМ	Количество полученных первичных баллов	Процент участников экзамена в субъекте Российской Федерации, получивших соответствующий первичный балл за выполнения задания в группах участников экзамена с разными уровнями подготовки			
		в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б., %	в группе от 61 до 80 т.б., %	в группе от 81 до 100 т.б., %
1-12	0	95,76	55,79	21,72	5,5
1-12	1	4,24	44,21	78,28	94,5
1-13	0	95,15	68,18	22,95	4,59
1-13	1	4,85	31,82	77,05	95,41
1-14	0	98,18	88,02	43,44	11,93
1-14	1	1,82	11,98	56,56	88,07
1-15	0	97,58	85,12	47,54	11,93
1-15	1	2,42	14,88	52,46	88,07
1-16	0	95,15	48,76	11,48	0,92
1-16	1	4,85	51,24	88,52	99,08
1-17	0	100	93,39	48,77	3,67
1-17	1	0	6,61	51,23	96,33
1-18	0	94,55	61,57	20,9	8,26
1-18	1	5,45	38,43	79,1	91,74
1-19	0	92,73	60,74	17,62	1,83
1-19	1	7,27	39,26	82,38	98,17
1-20	0	99,39	73,97	13,93	0,92
1-20	1	0,61	26,03	86,07	99,08
1-21	0	98,79	72,73	15,16	2,75
1-21	1	1,21	27,27	84,84	97,25
1-22	0	96,36	79,75	42,21	10,09
1-22	1	3,64	20,25	57,79	89,91
1-23	0	99,39	95,45	52,05	8,26

Номер задания / критерия оценивания в КИМ	Количество полученных первичных баллов	Процент участников экзамена в субъекте Российской Федерации, получивших соответствующий первичный балл за выполнения задания в группах участников экзамена с разными уровнями подготовки			
		в группе не преодолевших минимальный балл, %	в группе от минимального до 60 т.б., %	в группе от 61 до 80 т.б., %	в группе от 81 до 100 т.б., %
1-23	1	0,61	4,55	47,95	91,74
1-24	0	100	100	99,18	60,55
1-24	1	0	0	0,82	39,45
1-25	0	100	100	92,62	42,2
1-25	1	0	0	7,38	57,8
1-26	0	100	100	97,54	56,88
1-26	1	0	0	1,64	17,43
1-26	2	0	0	0,82	25,69
1-27	0	100	100	94,67	30,28
1-27	1	0	0	3,69	22,02
1-27	2	0	0	1,64	47,71

Выявление для каждой группы участников ЕГЭ заданий, выполняемых с наибольшей и наименьшей успешностью –

Таблица 2-15

	в группе не преодолевших минимальный балл	в группе от минимального до 60 т.б.	в группе от 61 до 80 т.б.	в группе от 81 до 100 т.б.
1	2	3	4	5
Наиболее успешно выполненные задания **базового** уровня сложности	№1 (Умение представлять и считывать данные в разных типах информационны...) – 62.4%; №10	№1 (Умение представлять и считывать данные в разных типах информационны...) – 93.5%; №10 (Информационный поиск		

	(Информационный поиск средствами текстового процессора) – 50.6%	средствами текстового процессора) – 79.8%		
Наименее успешно выполненные **задания** базового уровня сложности	№8 (Знание основных понятий и методов) – 1.8%; №9 (Умение обрабатывать числовую информацию в электронных таблицах) – 3.5%	№9 (Умение обрабатывать числовую информацию в электронных таблицах) – 20.6%; №8 (Знание основных понятий и методов) – 35.5%	№9 (Умение обрабатывать числовую информацию в электронных таблицах) – 64.0%; №8 (Знание основных понятий и методов) – 75.7%	
Наиболее успешно выполненные задания **повышенного** уровня сложности	№18 (Умение использовать электронные таблицы для обработки целочисленных...) – 5.3%; №13 (Умение использовать маску подсети) – 4.7%	№16 (Вычисление рекуррентных выражений) – 51.6%; №12 (Умение исполнить алгоритм для конкретного исполнителя с фиксированн...) – 43.5%	№16 (Вычисление рекуррентных выражений) – 88.7%; №20 (Умение найти выигрышную стратегию игры) – 85.8%	№16 (Вычисление рекуррентных выражений) – 99.1%; №20 (Умение найти выигрышную стратегию игры) – 98.3%
Наименее успешно выполненные задания **повышенного** уровня сложности		№23 (Умение анализировать ход исполнения алгоритма) – 4.4%; №17 (Умение составить алгоритм обработки числовой последовательности и з...) – 6.5%	№23 (Умение анализировать ход исполнения алгоритма) – 48.6%; №17 (Умение составить алгоритм обработки числовой последовательности и з...) – 51.0%	№11 (Умение подсчитывать информационный объём сообщения) – 87.0%; №15 (Знание основных понятий и законов математической логики) – 88.7%
Наиболее успешно выполненные задания **высокого** уровня сложности		№21 (Умение построить дерево игры по заданному алгоритму и найти выигрыш...) – 27.0%; №24 (Умение создавать	№21 (Умение построить дерево игры по заданному алгоритму и найти выигрыш...) – 85.0%; №25 (Умение создавать	№21 (Умение построить дерево игры по заданному алгоритму и найти выигрыш...) – 97.4%; №25 (Умение создавать

		собственные программы) – 0.0%	собственные программы) – 7.3%	собственные программы) – 59.1%
Наименее успешно выполненные задания **высокого** уровня сложности			№24 (Умение создавать собственные программы) – 0.8%; №26 (Умение обрабатывать целочисленную информацию с использованием сорти...) – 1.6%	№26 (Умение обрабатывать целочисленную информацию с использованием сорти...) – 34.4%; №24 (Умение создавать собственные программы) – 40.9%

3.1.2. Прочие результаты статистического анализа (при наличии)

По результатам статистического анализа выполнения заданий КИМ ЕГЭ по информатике в 2026 году выявлено, что наибольший процент выполнения (свыше 80%) наблюдается по заданиям базового уровня №1 (86,4%), №2 (73,3%), №4 (81,9%). Наименьший процент выполнения (менее 10%) – по заданиям №24 (4,0%), №26 (8,8%), №27 (4,7%), что характерно для заданий высокого уровня сложности, требующих навыков программирования и анализа данных. Средний процент выполнения по всем заданиям составил 42,3%.

Можно выделить задания с наименьшим процентом выполнения (независимо от групп с набранными баллами) по уровням сложности:

1. Базовый уровень – 4, 8, 9 задания.
2. Повышенный уровень – 11, 17, 23 задания.
3. Высокий уровень сложности – 24, 26, 27 задания.

Далее в качестве примера приведены задания из открытого варианта 313

Пример задания 4:

Задание 4

По каналу связи передаются сообщения, содержащие только буквы из набора: А, Д, К, Н, Р. Для передачи используется двоичный код, удовлетворяющий условию Фано. Кодовые слова для некоторых букв известны: Р – 0101, Н – 011. Для трёх оставшихся букв А, К и Д кодовые слова неизвестны. Какое количество двоичных знаков требуется для кодирования слова КАНАДКА, если известно, что оно закодировано **минимально** возможным количеством двоичных знаков?

Примечание. Условие Фано означает, что никакое кодовое слово не является началом другого кодового слова. Это обеспечивает возможность однозначной расшифровки закодированных сообщений.

Возможное решение задания 4:
Частоты в слове КАНАДКА: А(3), К(2), Н(1), Д(1).
Оптимальные коды (Фано): А=1 (1 бит), К=00 (2), Н=011 (3, фикс.), Д=0100 (4).
Проверка: ни один код не является префиксом другого.
Итог: 3*1 + 2*2 + 1*3 + 1*4 = 14 бит.

Пример задания 8:

Задание 8

Все пятибуквенные слова, составленные из букв А, К, Ц, Е, Н, Т, записаны в алфавитном порядке и пронумерованы.
Вот начало списка:
1. ААААА
2. ААААЕ
3. ААААК
4. ААААН
5. ААААТ
6. АААЦ
……
Определите, под каким номером в этом списке стоит первое слово с чётным номером, которое не начинается с букв А, Е или К и при этом содержит в своей записи не менее двух букв Ц.

Примечание. Слово – последовательность идущих подряд букв, не обязательно осмысленная.

Возможное решение задания 8 (программное):

```
letters = ['А','Е','К','Н','Т','Ц']
for n in range(6**5):
    t = n; w = []
    for _ in range(5):
        w.append(letters[t % 6])
        t //= 6
    w = ''.join(reversed(w))
    pos = n + 1
    if pos % 2 == 0 and w[0] not in 'АЕК' and w.count('Ц') >= 2:
        print(pos, w)
        break
```

Результат: слово НААЦЦ на позиции 3924.

Пример задания 9:

Задание 9

Задание выполняется с использованием прилагаемых файлов.

Откройте файл электронной таблицы, содержащей в каждой строке пять натуральных чисел. Определите сумму чисел в строке таблицы с **наименьшим** номером, содержащей числа, для которых выполнены оба условия:

– в строке все числа различны;

– удвоенная сумма максимального и минимального чисел строки больше суммы оставшихся трёх её чисел.

В ответе запишите только число.

Возможное решение задания 9 (программное):

```
import csv
with open('9.csv') as f:
    for i, row in enumerate(csv.reader(f), 1):
        nums = list(map(int, row))
        if len(set(nums)) != 5:
            continue
        mx, mn = max(nums), min(nums)
        if 2*(mx+mn) > sum(nums)-mx-mn:
            print(sum(nums));
            break
```

Пример задания 11:

Задание 11

На предприятии каждой изготовленной детали присваивают серийный номер, состоящий из 172 символов. В базе данных для хранения каждого серийного номера отведено одинаковое и минимально возможное целое число байт. При этом используется посимвольное кодирование серийных номеров, все символы кодируются одинаковым и минимально возможным числом бит. Известно, что для хранения 187 564 серийных номеров отведено не более 39 Мбайт памяти. Определите максимально возможную мощность алфавита, используемого для записи серийных номеров. В ответе запишите только целое число.

Пример задания 17:

Задание 17

Задание выполняется с использованием прилагаемых файлов.

В файле содержится последовательность целых чисел. Её элементы могут принимать целые значения от –100 000 до 100 000 включительно. Определите количество пар последовательности, в которых элементы не равны, а абсолютное значение их разности делится на минимальный положительный элемент последовательности, кратный 33. Гарантируется, что такой элемент в последовательности есть. В ответе запишите количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Возможное решение задания 11:

39 МБ = 39*2^20 = 40 894 464 байта.
Байт на номер <= 40 894 464 / 187 564 = 218.05 -> макс. 218 байт/номер.
218*8 = 1744 бит/номер. 1744 / 172 = 10.14 -> макс. 10 бит/символ.
Мощность алфавита = 2^10 = 1024.
Ответ 1024

Пример задания 17:

Задание 17

Задание выполняется с использованием прилагаемых файлов.

В файле содержится последовательность целых чисел. Её элементы могут принимать целые значения от –100 000 до 100 000 включительно. Определите количество пар последовательности, в которых элементы не равны, а абсолютное значение их разности делится на минимальный положительный элемент последовательности, кратный 33. Гарантируется, что такой элемент в последовательности есть. В ответе запишите количество найденных пар, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два идущих подряд элемента последовательности.

Возможное решение задания 17 (программное):

```
with open('17.txt') as f:
    data = list(map(int, f.readlines()))
min33 = min(x for x in data if x > 0 and x % 33 == 0)
cnt = 0
max_s = -10**9
for i in range(len(data)-1):
    a, b = data[i], data[i+1]
    if a != b and abs(a-b) % min33 == 0:
        cnt += 1
        max_s = max(max_s, a+b)
print(cnt, max_s)
```

Пример задания 23:

Задание 23

Исполнитель преобразует число на экране.
У исполнителя есть две команды, которые обозначены латинскими буквами:

***A.* Прибавь 1**

***B.* Поменяй местами**

Первая из этих команд увеличивает число на экране на 1. Вторая команда применяется только к числу, у которого цифра в разряде десятков по значению меньше цифры, стоящей в разряде единиц, и действует, заменяя число на экране числом, в котором цифры двух младших разрядов поменялись местами.
Программа для исполнителя – это последовательность команд.
Сколько существует программ, для которых при исходном числе 100 результатом является число 150?
Траектория вычислений программы – это последовательность результатов выполнения всех команд программы.
Например, для программы ***ABA*** при исходном числе 13 траектория состоит из чисел 14, 41, 42.

Возможное решение задания 23 (программное):

```
def swap(n):
    t = (n//10)%10
    u = n%10
    return None if t >= u else (n//100)*100 + u*10 + t
dp = [0]*151
dp[100] = 1
for cur in range(100, 150):
    if dp[cur]:
        if cur+1 <= 150:
            dp[cur+1] += dp[cur]
        s = swap(cur)
        if s and s <= 150:
            dp[s] += dp[cur]
print(dp[150])
```

Пример задания 24:

Задание 24

Задание выполняется с использованием прилагаемых файлов.

Текстовый файл состоит из цифр 0, 1, 2, 3, 4 и знаков арифметических операций «–» и «*» (вычитание и умножение). Определите максимальное количество символов в непрерывной последовательности, которая является корректным арифметическим выражением с целыми неотрицательными числами. В этом выражении никакие два знака арифметических операций не стоят рядом, в записи чисел отсутствуют незначащие (ведущие) нули и число 0 не имеет знака.
В ответе укажите количество символов.

Возможное решение задания 24 (программное):

```
with open('24.txt') as f:
    s = f.read().strip()
n = len(s)
max_len = 0
i = 0
while i < n:
    if s[i] not in '01234':
        i += 1
        continue
    start = i
    expect_num = True
    j = i
    ok = True
    while j < n and ok:
        if expect_num:
            if s[j] not in '01234':
                ok = False; break
            if s[j]=='0' and j+1<n and s[j+1] in '01234':
                ok = False; break
            while j < n and s[j] in '01234': j += 1
            expect_num = False
        else:
            if s[j] not in '-*':
                ok = False; break
            j += 1
            expect_num = True
    if ok and not expect_num:
        max_len = max(max_len, j-start)
    i = start + 1
print(max_len)
```

Пример задания 26:

Задание 26

Задание выполняется с использованием прилагаемых файлов.

Сервер выполняет запросы на передачу данных, при этом сведения о каждом выполненном запросе (время регистрации, идентификатор клиента и объём переданных данных) сохраняются в журнале работы, а сам запрос – в специальном разделе памяти сервера, имеющем ограниченный объём. Каждый раз, когда в специальном разделе остаётся недостаточно свободной памяти, сервер создаёт резервную копию всех накопленных там данных, после чего освобождает раздел и продолжает выполнение запросов. Напишите программу для обработки журнала работы сервера и с её помощью определите идентификатор клиентского устройства, с которого на сервер был передан наибольший суммарный объём данных не позднее 11:59:59, а также сумму объёмов двух наибольших резервных копий специального раздела (в Кбайт).

Входные данные

Первая строка входного файла (журнала работы сервера) содержит два натуральных числа: N ($N < 1\,000\,000$) – количество строк в журнале и K ($K < 1\,000\,000$) – вместимость специального раздела памяти сервера в Кбайт. Каждая из следующих N строк содержит информацию об одном выполненном запросе: время регистрации запроса в формате ЧЧ:ММ:СС (часы, минуты, секунды) и два натуральных числа: C ($C < 1\,000\,000$) – идентификатор клиентского устройства и S ($S < K$) – объём данных запроса в Кбайт.

Выходные данные

В ответе запишите два числа: сначала идентификатор устройства, с которого на сервер был передан наибольший суммарный объём данных не позднее 11:59:59, а затем сумму объёмов двух наибольших резервных копий специального раздела (в Кбайт).

Типовой пример организации данных во входном файле

8 140000
01:01:01 101 20000
03:03:03 202 110000
05:05:05 101 90000
07:07:07 303 62000
10:10:10 101 48000
15:15:15 202 12000
21:21:21 303 120000
23:23:23 404 134000

При таких исходных данных резервное копирование специального раздела выполняется четыре раза: в 05:05:05 (в объёме 130 000 Кбайт), в 07:07:07 (в объёме 90 000 Кбайт), в 21:21:21 (в объёме 122 000 Кбайт) и в 23:23:23 (в объёме 120 000 Кбайт). Всего на сервер передано 596 000 Кбайт данных: 158 000, 122 000, 182 000 и 134 000 Кбайт от клиентов с идентификаторами 101, 202, 303 и 404 соответственно. Ответ для приведённого примера: 101 252 000.

Типовой пример имеет иллюстративный характер. Для выполнения задания используйте данные из прилагаемого файла.

Возможное решение задания 26 (программное):

```
with open('26.txt') as f:
    N, K = map(int, f.readline().split())
    client_total = {}
    backups = []
    cur = 0
    for _ in range(N):
        t, cid, s = f.readline().split()
        cid = int(cid); s = int(s)
        if t <= '11:59:59':
            client_total[cid] = client_total.get(cid,0) + s
        if cur + s > K:
            backups.append(cur)
            cur = 0
        cur += s
    if cur:
        backups.append(cur)
max_id = max(client_total, key=client_total.get)
backups.sort(reverse=True)
print(max_id, backups[0] + backups[1])
```

Пример задания 27:

Задание 27

Задание выполняется с использованием прилагаемых файлов.

Фрагмент звёздного неба спроецирован на плоскость с декартовой системой координат. Учёный решил провести кластеризацию полученных точек, являющихся изображениями звёзд, то есть разбить их множество на N непересекающихся непустых подмножеств (кластеров), таких, что точки каждого подмножества лежат внутри прямоугольника со сторонами длиной H и W, причём эти прямоугольники между собой не пересекаются. Стороны прямоугольников не обязательно параллельны координатным осям. Гарантируется, что такое разбиение существует и единственно для заданных размеров прямоугольников.

Будем называть центром кластера точку (звезду) этого кластера, сумма расстояний от которой до всех остальных его точек минимальна. Для каждого кластера гарантируется единственность его центра. Расстояние между двумя точками на плоскости $A(x_1, y_1)$ и $B(x_2, y_2)$ вычисляется по формуле:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Каждая звезда помимо координат на плоской карте характеризуется своим спектральным классом и классом светимости. Спектральный класс определяет цвет (который связан с температурой звезды) следующим образом.

Обозначение спектрального класса (латинская буква)	O	B	A	F	G	K	M
Цвет звезды	Голубой	Бело-голубой	Белый	Жёлто-белый	Жёлтый	Оранже-вый	Красный

Каждый из спектральных классов, в свою очередь, делится на подклассы от 0 до 9 в порядке уменьшения температуры. Обозначение подкласса ставится после обозначения спектрального класса (например, B2).

Класс светимости звезды обозначим римскими цифрами от I до VII.

Обозначение класса светимости	I	II	III	IV	V	VI	VII
Светимость	Сверх-гигант	Яркий гигант	Гигант	Суб-гигант	Карлик	Суб-карлик	Белый карлик

В файле А хранится информация о точках **двух** кластеров, где $H = 6{,}0$ и $W = 5{,}5$ для каждого кластера. В каждой строке сначала записана информация о расположении на карте одной звезды: координата x и координата y. Далее в той же строке для звёзд классов светимости I–VI указываются спектральный класс, подкласс и класс светимости. Обозначения классов ничем не разделяются. Для звёзд класса светимости VII (Белый карлик) обозначения спектрального класса и подкласса в файле не указываются. Известно, что количество точек не превышает 2000.

В файле Б хранятся координаты точек **трёх** кластеров, где $H = 6{,}0$ и $W = 5{,}5$ для каждого кластера. Известно, что количество точек не превышает 10 000. Структура хранения информации в файле Б аналогична структуре в файле А.

Для файла А определите координаты центра каждого кластера, затем найдите два числа: A_1 – количество красных гигантов, расположенных на расстоянии не более 1,0 от центра кластера, который содержит наименьшее количество точек, и A_2 – расстояние между центрами кластеров.
Для файла Б определите координаты центра каждого кластера, затем найдите два числа: B_1 – наименьшее расстояние от центра кластера до голубого гиганта, принадлежащего к этому кластеру, и B_2 – наибольшее количество жёлтых карликов в одном кластере.

В ответе запишите четыре числа: в первой строке – сначала A_1, затем целую часть произведения $A_2 \times 10\,000$; во второй строке – сначала целую часть произведения $B_1 \times 10\,000$, затем B_2.

Пример организации данных в одном из исходных файлов для случая четырёх звёзд

5,01788	*8,32466*	*G2V*
4,289251	*6,955186*	*VII*
4,619358	*5,524697*	*B7V*
6,91934	*20,425391*	*G2V*

Внимание! Пример приведён в иллюстративных целях для произвольных значений, не имеющих отношения к заданию. Для выполнения задания используйте данные из прилагаемых файлов.

Возможное решение задания 27 (программное):

```
import math
import sys

def parse_star(line):
    parts = line.strip().split()
    x = float(parts[0].replace(',', '.'))
    y = float(parts[1].replace(',', '.'))
    cls = parts[2] if len(parts) > 2 else 'VII'
    return x, y, cls

def dist(a, b):
    return math.hypot(a[0] - b[0], a[1] - b[1])
```

```python
def cluster_by_mst(coords, k):
    m = len(coords)
    edges = [(dist(coords[i], coords[j]), i, j) for i in range(m) for j in range(i + 1, m)]
    edges.sort(key=lambda x: x[0])
    parent = list(range(m))
    def find(x):
        while parent[x] != x:
            parent[x] = parent[parent[x]]
            x = parent[x]
        return x
    def union(x, y):
        rx, ry = find(x), find(y)
        if rx != ry:
            parent[ry] = rx
            return True
        return False
    comps = m
    for _, i, j in edges:
        if union(i, j):
            comps -= 1
            if comps == k:
                break
    cl = {}
    for i in range(m):
        cl.setdefault(find(i), []).append(i)
    return list(cl.values())

def find_center_idx(points, coords):
    best_i = 0
    best_s = float('inf')
    for i, p in enumerate(points):
        s = sum(dist(coords[p], coords[q]) for q in points)
        if s < best_s:
            best_s = s
            best_i = i
    return points[best_i]
```

```
def solve_a(path):
    stars = [parse_star(l) for l in open(path, encoding='utf-8')]
    coords = [(x, y) for x, y, _ in stars]
    clusters = cluster_by_mst(coords, 2)

    cl_data = [(len(c), c) for c in clusters]
    cl_data.sort(key=lambda x: x[0])
    small = cl_data[0][1]
    large = cl_data[1][1]

    small_center = find_center_idx(small, coords)
    large_center = find_center_idx(large, coords)

    a1 = 0
    for idx in small:
        _, _, cls = stars[idx]
        if not cls.startswith('M'):
            continue
        lum = cls[2:]
        if lum in ('I', 'II', 'III'):
            if dist(coords[small_center], coords[idx]) <= 1.0:
                a1 += 1
    a2 = dist(coords[small_center], coords[large_center])
    return a1, a2
```

```
def solve_b(path):
    stars = [parse_star(l) for l in open(path, encoding='utf-8')]
    coords = [(x, y) for x, y, _ in stars]
    clusters = cluster_by_mst(coords, 3)

    b1 = float('inf')
    b2 = 0
    for cl in clusters:
        cent = find_center_idx(cl, coords)
        for idx in cl:
            _, _, cls = stars[idx]
            if len(cls) >= 3 and cls[0] in 'OB' and cls[2:] in ('I', 'II'):
                dval = dist(coords[cent], coords[idx])
                if dval < b1:
                    b1 = dval
        yellow = sum(1 for idx in cl if stars[idx][2].startswith('G') and stars[idx][2][2:] == 'V')
        if yellow > b2:
            b2 = yellow
    return b1, b2

if __name__ == '__main__':
    a1, a2 = solve_a(sys.argv[1])
    b1, b2 = solve_b(sys.argv[2])
    print(a1)
    print(int(a2 * 10000))
```

3.2. Методический анализ выполнения заданий КИМ обучающимися с разными уровнями подготовки и рекомендации для учителей по совершенствованию преподавания учебного предмета

3.2.1. Методический анализ выполнения заданий обучающимися с минимальным уровнем подготовки (не преодолевших минимальный балл), включая методические рекомендации для учителей

3.2.1.1. Описание предметной подготовки участников ЕГЭ с минимальным уровнем подготовки, анализ типичных дефицитов в подготовке

- Описание достигнутых предметных результатов ФГОС: освоенных тем программы, сформированных умений и т.п. (если имеются)

Участники с минимальным уровнем подготовки (не преодолевшие минимальный балл) наиболее успешно справились с заданиями №1 (представление данных в информационных моделях) – 38,18% выполнения, №4 (кодирование/декодирование информации) – 27,27%, №10 (информационный поиск) – 51,52%. Также относительно успешно выполнены задания №2 (таблицы истинности) – 27,27% и №3 (поиск в реляционных БД) – 27,27%. Это свидетельствует о частичной сформированности базовых навыков работы с информацией.

- Анализ типичных дефицитов в предметной подготовке: неосвоенные или слабо освоенные темы программы, несформированные или недостаточно сформированные умения, навыки, виды познавательной деятельности, проявляющиеся в типичных ошибках при выполнении соответствующих заданий

Таблица 2-16

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП[2]
1.	Формальное исполнение алгоритмов (задания №5, 6, 12)	8–11 кл.
2.	Измерение количества информации, кодирование (задания №7, 8, 11)	7, 10–11 кл.
3.	Позиционные системы счисления (задание №14)	8, 10 кл.

[2] Здесь и далее: См. пункт 3 Спецификации КИМ ЕГЭ 2026 года по учебному предмету.

4.	Основы алгебры логики (задание №15)	8, 10 кл.
5.	Обработка числовой информации в электронных таблицах (задание №9)	9, 11 кл.
6.	Программирование: создание программ (задания №24–27)	8–11 кл.

Типичные ошибки:

- задание 5 (успешность выполнения 4,71%) – ошибки при переводе из одной системы счисления в другую, неумение выполнить пошаговый алгоритм; отсутствие ответа.
- задание 6 (успешность выполнения 14,71%) – ошибки при определении результата выполнения алгоритма исполнителем Черепаха, некоторые обучающиеся перепутали значения команд «вперед – назад», «направо – налево»; отсутствие ответа.
- задание 7 (успешность выполнения 19,41%) – ошибки при определении объёма памяти для хранения звуковой информации, при переводе единиц информации; применении формул, связывающих частоту дискретизации, количество информации и время; отсутствие ответа.
- задание 8 (успешность выполнения 1,76%) – ошибки в использовании методов измерения количества информации; отдельные обучающиеся пытались выполнить задание методом подбора; отсутствие ответа.
- задание 11 (успешность выполнения 2,94%) – ошибки при подсчете информационного объёма сообщения, определении взаимосвязи между числом серийных номеров и объемом памяти в базе данных; вычислительные ошибки; отсутствие ответа.
- задание 14 (успешность выполнения 1,76%) – ошибки в нахождении значения выражения в пятеричной системе счисления, при переводе чисел из одной системы счисления в другую; отсутствие ответа.
- задание 15 (успешность выполнения 2,35%) - ошибки при выполнении логических операций, связанные с незнанием основных понятий и законов математической логики; отсутствие ответа.
- задание 9 (успешность выполнения 3,53%) – ошибки при обработке числовой информации в электронных таблицах; обучающиеся учитывали только одно условие из двух; отсутствие ответа.
- задания 24 – 27 (успешность выполнения 0%) – ошибки при создании программ, связанные с неумением составлять программы, незнанием методов обработки символьной информации, целочисленной информации, анализа данных; отсутствие ответа.

- Реалистичные «зоны роста» в части предметной подготовки для уверенного преодоления минимального балла ЕГЭ

Для уверенного преодоления минимального балла участникам данной группы необходимо:

- освоить базовые навыки формального исполнения алгоритмов (задания №5, №6);
- научиться определять объём памяти для хранения графической и звуковой информации (задание №7);
- овладеть навыками подсчёта информационного объёма сообщения (задание №11);
- улучшить знания позиционных систем счисления (задание №14) и основ математической логики (задание №15).

Рекомендуется сконцентрироваться на темах раздела «Теоретические основы информатики» и базовых алгоритмических конструкциях.

3.2.1.2. Методический анализ качества освоения метапредметных результатов ФГОС участниками ЕГЭ с минимальным уровнем подготовки

- Развернутый комментарий на основе заданий / групп заданий, на успешность выполнения которых могла повлиять достаточная или недостаточная сформированность метапредметных умений, с указанием соответствующих метапредметных умений и их проявления в типичных ошибках в ответах участников экзамена на соответствующие задания;

Допущенные обучающимися данной группы типичные ошибки, рассмотренные выше, позволяют сделать выводы относительно недостаточной сформированности УУД, повлиявшей на успешность выполнения заданий КИМ.

У участников с минимальным уровнем подготовки недостаточно сформированы познавательные УУД, связанные с анализом информации и логическими рассуждениями: выявлять закономерности и противоречия в рассматриваемых явлениях; устанавливать существенный признак или основания для сравнения, классификации и обобщения;

Отсутствует способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; обучающиеся не владеют научной терминологией, ключевыми понятиями и методами; не умеют анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях.

При работе с информацией испытывают существенные трудности при создании текстов в различных форматах с учетом назначения информации, выборе оптимальной формы представления и визуализации.

При выполнении заданий №5, №6 проявляется неумение формально исполнять алгоритмы, что требует развития алгоритмического мышления. Задания №24 – 27 (программирование) практически не выполняются данной группой, что свидетельствует о полном отсутствии навыков программирования.

Регулятивные УУД не сформированы в части самоорганизации: самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи; самостоятельно составлять план решения проблемы с учетом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение. Навыки самоконтроля - вносить коррективы в деятельность, оценивать соответствие результатов целям – также не сформированы.

- Предположение о достигнутых и недостигнутых метапредметных результатах ФГОС

Метапредметные результаты ФГОС у группы с минимальным уровнем подготовки достигнуты лишь частично. Удовлетворительно сформированы отдельные базовые логические действия (умение работать с таблицами, схемами). Недостаточно сформированы: знаково-символические действия (кодирование/декодирование), навыки смыслового чтения (понимание условия задачи), регулятивные действия (самопроверка, коррекция ошибок). Требуется систематическая работа по формированию всех групп УУД.

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

Общие рекомендации:

1. Проводить диагностические работы в начале 10 класса для выявления группы риска.
2. Организовать регулярные тренировочные упражнения на формальное исполнение алгоритмов (задания №5, №6, №12) с пошаговым разбором.
3. Уделять особое внимание изучению систем счисления (задание №14) и основ алгебры логики (№15).
4. Использовать на уроках задания на вычисление информационного объёма (№7, №11).
5. Развивать читательскую грамотность через анализ условия задачи, выделение ключевых данных.
6. Применять чек-листы самопроверки для формирования регулятивных УУД.

Приведем конкретные примеры методических приемов и заданий для обучающихся данной группы по трудным аспектам.

Для слабых учеников главная проблема при выполнении заданий на формальное исполнение алгоритмов - это попытка выполнить всё «в уме» и потеря переменной из вида уже на втором шаге.

Алгоритмы разбора должны быть жесткими, почти механическими. Ученик должен воспринимать их не как математику, а как инструкцию по заполнению анкеты или игру в «угадайку» с компьютером.

Ниже представлены три пошаговых алгоритма для разных типов заданий.

Алгоритм №1: «Строгий конвой» (для линейных алгоритмов и присваиваний)

Тип задания: Дана последовательность команд, нужно найти значение переменной. Проблема ученика: Он путает a = a + b (сложить) и a = b (заменить).

Пошаговый регламент («Закон таблицы»):

1. Правило одной строки. Никогда не смотри на следующую строчку, пока полностью не заполнил текущую строку таблицы.
2. Выдели изменяемое. Перед выполнением команды найди в таблице ячейку, которая стоит слева от знака :=. Это единственное место, куда мы сейчас что-то запишем.

3. Сборка правой части (Эффект закрытой крышки).

- Закрой пальцем правую часть уравнения.
- Найди все переменные справа.
- Подставь значения ТОЛЬКО ИЗ ПРЕДЫДУЩЕЙ СТРОКИ таблицы. Не из этой! Из предыдущей!
- Вычисли результат на калькуляторе или черновике.

4. Запись. Запиши полученное число в колонку той переменной, которую выделил в пункте 2.

5. Проверка дублирования. Если команда выглядит как x := x, просто перепиши число из ячейки выше в эту же ячейку. Ничего не меняется.

Пример-шпаргалка для ученика:

Шаг	Команда	x	y
0	Начало	**5**	**10**
1	`x := x + y`	$5 + 10 = \mathbf{15}$	10
2	`y := x - y`	15	$15 - 10 = \mathbf{5}$
3	`x := x - y`	$15 - 5 = \mathbf{10}$	5

Алгоритм №2: «Дерево ветвлений» (для условного оператора IF)

Тип задания: Определить итоговое значение или количество путей.

Проблема ученика: Выполнение обеих веток (if и else) сразу или пропуск ветки, если условие ложно.

Пошаговый регламент («Метод светофора»):

1. Выполни базу. Сначала посчитай всё, что идет до слова IF. Заполни таблицу значений до этого момента.
2. Красный/Зеленый. Прочитай условие после IF.

Если там знак < или >, подставь числа из таблицы и проверь.

Напиши рядом со словом IF маркером крупно: ДА или НЕТ.

3. Режим «Слепая зона».

Если написано НЕТ: весь блок кода внутри фигурных скобок (или под BEGIN...END) становится невидимым. Мы его даже не читаем. Просто переходим к строке ПОСЛЕ всего блока IF.

Если написано ДА: вычеркни условие. Выполняй только то, что осталось внутри блока.

4. Вложенность (Матрешка). Если внутри первого IF есть второй IF, вернись к пункту №2. Снова проверяй условие только текущей «матрешки».

Совет для учителя: Давайте задачи, где во всех вариантах ответов даны разные результаты для ветки ELSE. Слабый ученик быстро поймет: чтобы получить правильный ответ, ему обязательно нужно проверить, работает ли ветка «иначе».

Алгоритм №3: «Циклический конвейер» (FOR / WHILE)

Тип задания: Сколько раз выполнится цикл или какое будет последнее значение.

Проблема ученика: Бесконечный цикл в голове или непонимание, когда счетчик меняется.

Пошаговый регламент («Таблица итераций»): Для слабых учеников циклы - самое сложное. Таблица должна иметь специфический вид:

1. Подготовка (До цикла):

- Запиши начальные значения переменных.
- Для цикла for i := 1 to N выпиши себе отдельно список значений i: 1, 2, 3 ... N.

2. Тело цикла (Главный столбец).

- Создай в тетради столбец «Результат тела».
- Возьми первое значение i (например, 1).
- Проведи вычисления внутри цикла, используя это i. Запиши, чему стали равны другие переменные.

3. Конец итерации.

- Посмотри на условие выхода (while или границы for).
- Если условие позволяет идти дальше, бери следующее i.
- Важно: При новом заходе в цикл НЕ МЕНЯЙ те переменные, которые меняются только внутри тела, обратно! Меняй только счетчик i.

Пример (цикл While):

```
q := 1; k := 1; while q < 10 do begin q := q + k; k := k + 1; end;
```

Номер круга	Значение q (на входе)	Значение k (на входе)	Проверка условия q < 10	Действие q := q+k	Действие k := k+1
Старт	1	1	$1 < 10$ (Да)	$1 + 1 = 2$	$1 + 1 = 2$
Круг 1	2	2	$2 < 10$ (Да)	$2 + 2 = 4$	$2 + 1 = 3$
Круг 2	4	3	$4 < 10$ (Да)	$4 + 3 = 7$	$3 + 1 = 4$
Круг 3	7	4	$7 < 10$ (Да)	$7 + 4 = 11$	$4 + 1 = 5$
Круг 4	11	5	$11 < 10$ (**Нет**)	**СТОП**	

Финальный результат: q = 11, k = 5.

Общие методические правила:

- Запрет на сокращения. Ученик не имеет права написать x=15. Он должен написать процесс: 5 + 10 = 15.
- Цветовая маркировка. Пусть ученики выделяют переменную, которая меняется в данной строке, цветным карандашом. Это фокусирует внимание.
- «Глупый компьютер». Поиграйте в роли компьютера. Ученик дает вам команду x = x + 1, а вы намеренно делаете ошибку (например, умножаете), чтобы показать: машина делает ровно то, что написано, а не то, что вы имели в виду.

Для слабых учеников абстрактные понятия «бит», «логическое умножение» или «основание системы» звучат как иностранный язык. Чтобы они поняли материал, нужно превратить математику в игру с четкими правилами и визуальными опорами.

Вот три задания разного типа: детективное расследование, кулинарная логика и работа со шпионскими шифрами.

Задание 1. Детективное агентство: «Цветовой код доступа»

Цель: Понять перевод из двоичной системы в десятичную через физическое действие (включение/выключение).

Контекст: Ученики - специалисты по открыванию замков. Замок открывается только если набрать правильную сумму чисел на панели.

Оборудование: Карточки четырех цветов (Красный, Синий, Зеленый, Желтый). На каждой карточке с обратной стороны написано число:

Красный = 8

Синий = 4

Зеленый = 2

Желтый = 1 (Это веса разрядов для 4-битного числа).
Алгоритм выполнения:
Учитель называет «код»: например, 1011.
Ученики должны выложить карточки цветами вверх в том же порядке: Красный (есть), Синего (нет), Зеленого (есть), Желтого (есть).
Правило сейфа: Мы берем себе цифру на обороте ТОЛЬКО той карты, которая лежит лицом вверх («лампочка горит»).
В примере 1011 мы берем: 8 (красная) + 0 (синей нет) + 2 (зеленая) + 1 (желтая).
Сумма (8+2+1=11) — это пароль от замка.
Почему это работает для слабых учеников: Они не возводят двойку в степень. Они просто складывают физические предметы. Ошибка становится видна сразу: если сумма слишком маленькая, значит, забыли перевернуть нужную карту.

Задание 2. Кулинарная книга робота-повара (Основы алгебры логики)
Цель: Понять операции И (AND), ИЛИ (OR) и НЕ (NOT) на примере реальных условий.
Контекст: Робот умеет готовить блюдо, только если выполняются условия рецепта.
Вводные данные (Ингредиенты):
А: Есть мясо (Да/Нет)
Б: Есть картошка (Да/Нет)
В: Есть соль (Да/Нет)
Рецепты (Логические формулы): Ученикам дается таблица заказов. Им нужно поставить галочку «Готовить?», исходя из наличия продуктов.

Заказ	Формула (условие)	Склад: Мясо(А)=Да, Картошка(Б)=Нет, Соль(В)=Да	Результат
№1 Жаркое	`А И Б` (Есть мясо **И** есть картошка)	Да **И** Нет → **Ложь**	❌ Не готовим
№2 Суп	`А ИЛИ Б` (Есть мясо **ИЛИ** есть картошка)	Да **ИЛИ** Нет → **Истина**	✅ Готовим
№3 Салат	`НЕ В` (**НЕТ** соли)	**НЕ** Да → **Ложь**	❌ Не готовим
№4 Стейк	`(А И В) И (НЕ Б)`	(Да **И** Да) **И** (**НЕ** Нет) → Истина	✅ Готовим

Методическая подсказка: Используйте жесты. И - обе руки должны быть подняты. ИЛИ - достаточно поднять одну любую руку. НЕ - качаем головой «нет». Это кинестетически закрепляет смысл операций.

Творческое усложнение: Попросите учеников придумать свой рецепт для бургера, используя формулу: (Мясо ИЛИ Котлета) И Булка. Пусть проверят её на разных складах.

Задание 3. Шпионская станция: «Шифр шестнадцатеричного агента»

Цель: Научиться переводить числа из систем счисления > 10 и понять связь между двоичным кодом и текстом.

Проблема слабых учеников: Буквы вместо цифр пугают их.

Инструментарий: Таблица дешифратора (обязательно распечатать крупно):

- `10 = A`
- `11 = B`
- `12 = C`
- `13 = D`
- `14 = E`
- `15 = F`

Этап 1: Разведка (Перевод в Десятичную систему) Даны координаты в формате HEX (шестнадцатеричные): 2C, 1F, 1A.

Правило позиции: Вторая цифра справа - это «сколько полных пачек по 16».

Расшифровка 2C:

Смотрим в таблицу: C — это 12.

Считаем: (2 * 16) + 12.

32 + 12 = 44. Координата найдена.

Этап 2: Передача сообщения (ASCII / Двоичный код) Учитель объясняет: компьютер хранит буквы как числа. Нам повезло, у нас простой алфавит.

A = 1, B = 2, C = 3...

Но передает он их в двоичном виде!

Задание: Зашифруйте слово «ДА».

Переводим буквы в числа: Д — 5-я буква, А — 1-я.

Переводим эти числа в двоичную систему методом «деления пополам» (или через те же цветные карточки из Задания 1):

Число 5: одна восьмерка (8), ноль четверок (4), одна двойка (2), один туз (1). Итого: 101.

Число 1: нули и единица в конце: 001.

Финальный пакет данных: 101 001.

Проверка (Самоконтроль): Если ученик получил 101, он должен проверить себя обратным действием: 1 * 4 + 0 * 2 + 1 * 1 = 5. Пятая буква — Д. Верно.

Общие советы для этих заданий:

- Запрет калькуляторов. Для перевода маленьких чисел (до 31) мозги работают быстрее гаджетов.
- Магия числа 16. Постоянно повторяйте мантру: «В Нех живет 16 жильцов: 10 квартир занимают цифры, а еще 6 квартир отдают буквам A, B, C, D, E, F».

Для слабых учеников главная сложность в задачах на информационный объем - это путаница между количеством символов и весом одного символа, а также страх перед степенями двойки. Задачи должны опираться на наглядные образы: коробки, ячейки памяти и простые пропорции.

Вот три задания разного типа для 7–9 классов.

Задание 1. «Конвейерная лента» (Алфавитный подход)

Цель: Научить четко разделять мощность алфавита (N) и количество символов в тексте (K).

Контекст: Завод по упаковке подарков. Каждая коробка имеет ячейку с лампочкой. Лампочка может гореть разным цветом.

Условие: На складе есть ящики трех цветов: Красный, Синий, Зеленый. Это наш «алфавит». Робот-погрузчик составил сообщение из 20 ящиков: Красный-Красный-Зеленый-Синий... (всего 20 штук). Нужно найти вес сообщения.

Пошаговый алгоритм («Два вопроса»): Слабым ученикам нужно дать жесткую схему вопросов:

Вопрос А: Сколько всего разных знаков? (Мощность N)

Считаем пальцами: Красный (1), Синий (2), Зеленый (3).

Ответ: N = 3.

Вопрос Б: Какое наименьшее число бит нужно для одного знака? (i)

Здесь мы используем таблицу-опору (её лучше повесить на стену или вклеить в тетрадь):

Знаков (N)	Бит на знак (i)
1 – 2	1 бит
3 – 4	2 бита
5 – 8	3 бита

У нас 3 цвета. Смотрим в таблицу: попадает в диапазон «3–4». Значит, 1 ящик занимает 2 бита. (Важно объяснить: двумя битами можно закодировать 4 варианта, поэтому даже если у нас всего 3 цвета, мы всё равно тратим 2 бита на каждый).

Вопрос В: Сколько всего знаков в сообщении? (K)

Просто считаем длину строки. В условии сказано: 20 ящиков. K = 20.
Финал: Умножаем!
Объем = K * i = 20 * 2 = 40 бит.
Почему это работает: Мы не заставляем их считать логарифм $\log_2 3$. Мы заменяем вычисление визуальным поиском в таблице.

Задание 2. «Фотоальбом деда» (Единицы измерения и пропорции)
Цель: Закрепить разницу между Битами и Байтами, отработать перевод величин без сложных формул.
Контекст: Оцифровка старых фотографий.
Условие: Мы знаем, что маленькая картинка размером 10 * 10 пикселей весит 400 байт. Сколько будет весить такая же картинка (с такой же глубиной цвета), но размером 20 * 20?
Метод «Решетки» (Визуализация площади): Многие слабые ученики пытаются возвести числа в квадрат и сразу падают в математику. Нужно показать геометрию.
Рисуем первую картинку: Квадрат 10 * 10. Внутри него рисуем сетку. Всего клеточек: 10 * 10 = 100.
Рисуем вторую картинку: Квадрат 20 * 20. Клеточек: 20 * 20 = 400.
Сравниваем:
Вопрос: «Во сколько раз вторая картинка больше первой ПО ПЛОЩАДИ?»
Не делим 400 / 100 сразу. Сначала спросите: «А во сколько раз сторона стала длиннее?» (В 2 раза).
«Если сторона выросла в 2 раза, то площадь вырастет в 2 * 2 = 4 раза».
Вычисление:
Раз площадь выросла в 4 раза, значит, и информации стало в 4 раза больше.
Берем старый вес: 400 байт.
Новый вес: 400 * 4 = 1600 байт.
Перевод (если требуется в Кб):
Правило: «Чтобы получить Килобайты, подели байты на 1024». Но для слабых учеников часто достаточно правила «раздели примерно на 1000», либо задача ограничивается только байтами.
Ловушка: Если изменить размер до 30 * 10, площадь станет 300 (выросла в 3 раза), а не в 9 раз. Решетка помогает это увидеть.

Задание 3. «Музыкальная шкатулка» (Частота дискретизации и время)
Цель: Показать, как параметры звука влияют на объем файла.
Контекст: Запись голоса на диктофон.
Условие: Диктофон записывает звук. Мы точно знаем:
Глубина кодирования: 16 бит на один замер (это характеристика качества микрофона, она фиксирована).
Длительность: 1 секунда записи при частоте 1000 Гц весит 2000 байт.

Вопрос: Сколько места займет эта же запись, если мы увеличим частоту (четкость) в 2 раза (до 2000 Гц)?
Алгоритм «Рецепт пирога»: Объясняем файл как слоеный пирог:
Ширина слоя = Глубина (биты).
Длина слоев = Частота (сколько слоев в секунду).
Высота стопки = Время (секунды).
Анализируем условие:
Нам говорят, что 1 секунда при «скорости» 1000 весит 2000 байт.
Что изменилось?
Время осталось прежним (1 сек).
Качество (глубина) осталось прежним (16 бит).
Изменилась частота: была 1000, стала 2000. То есть частота увеличилась в 2 раза.
Логический вывод:
Если мы делаем замеры в 2 раза чаще, то за ту же секунду мы запишем в 2 раза больше данных.
Старый объем был 2000 байт.
Новый объем: 2000 * 2 = 4000 байт.
Проверка формулой (для сильных в классе, но слабый ученик просто верит пропорции): V = K * i * t
Вторая ситуация: V_2 = (2K) * i * t = 2 * V_1.

Общие методические советы:

- Магическое число 8. Постоянно повторяйте мантру: «8 бит = 1 байт». Если в задаче получилось 16 бит, всегда спрашивайте: «А сколько это байт? Надо разделить на 8».
- Избавляйтесь от приставок. Пока ученик не усвоил базовый расчет, прячьте слова «Килобайт» и «Мегабайт». Работайте только в Байтах и Битах. Приставки добавляйте на последнем этапе как простое деление столбиком.
- Блокнот-считалка. Выдайте ученикам полоску бумаги, где они будут выписывать:

```
[ ] N (разных значков)
[ ] i (бит на значек ИЗ ТАБЛИЦЫ)
[ ] K (всего значков)
[ ] V = K * i
```

Для слабых учеников самоконтроль - это не врожденное умение, а навык, который нужно формировать через внешние опоры. Главная проблема таких детей в том, что они пытаются «угадать» ответ или выполнить действие механически, не понимая его смысла.

Рекомендуем три методических приема для развития самоконтроля на уроках информатики (алгоритмизация, программирование и работа с информацией).

1. Прием «Светофор готовности» (Визуализация состояния)

Этот прием учит ученика вовремя замечать ступор и просить о помощи осознанно, а не когда уже всё сломано.

Инструментарий: У каждого ученика на краю парты лежат три карточки: зеленая, желтая, красная.

Механика работы:

Зеленый: «Я понимаю задание, двигаюсь сам».

Желтый: «У меня возник вопрос/ошибка, я жду своей очереди к учителю». Это сигнал: «Не мешайте мне думать еще минуту, я пытаюсь решить сам».

Красный: «Критическая ошибка / компьютер завис / я вообще не понимаю алгоритм, нужна немедленная помощь».

Как это развивает самоконтроль: Перед тем как поднять карточку, ученик вынужден задать себе вопрос: «На каком этапе я нахожусь? Я просто устал (желтый) или у меня реально ничего не загружается (красный)?». Это заменяет привычное махание рукой фразой: «Мое состояние сейчас «желтое», потому что я застрял на шаге №3 цикла».

2. Прием «Чек-лист пирата» (Алгоритмизированная проверка кода)

Слабые ученики боятся двоек. Чек-лист переводит фокус с оценки («учитель поставит два») на процесс сборки («сколько деталей конструктора я собрал»).

Инструментарий: Распечатанный список критериев перед началом практической работы.

Пример чек-листа для задания «Создание презентации»:

[ ] На первом слайде есть заголовок и имя автора.

[ ] Текст всех слайдов читаемый (шрифт 24 пт).

[ ] Все картинки имеют подпись снизу мелким шрифтом.

[ ] Настроена анимация смены слайдов (не вручную мышкой).

[ ] Файл сохранен в папке 7_Б_Фамилия под именем Проект.pptx.

Механика взаимопроверки: Ученики меняются местами. Задача проверяющего - не критиковать дизайн соседа, а просто поставить галочки в пустых квадратах. Если галочки нет, он пишет комментарий: «Здесь нет подписи к картинке льва».

Почему это работает: Оценка становится объективной. Ученик видит: «Я выполнил 4 из 5 пунктов». Это снижает тревожность и приучает проверять работу по пунктам, а не «на глаз».

3. Прием «Реверсивное ревью» (Проверка чужой ошибки по ролям)

Чтобы научиться находить свои баги, нужно сначала потренироваться на чужих. Слабый ученик часто не понимает, что именно искать.

Инструментарий: Готовый лист с кодом, содержащим намеренные ошибки (типичные для этого класса).

Разделение ролей (Игра): Класс делится на пары. Каждый получает строгую роль:

«Тестировщик»: Пытается «сломать» программу партнера. Нажимает все кнопки подряд, вводит буквы там, где нужны цифры. Цель - найти ошибку логики.

«Редактор»: Ищет только опечатки и грамматические ошибки в интерфейсе программы (например, слово «Парроль» вместо «Пароль»).

«Дизайнер»: Оценивает только выравнивание элементов и цвета.

Правило обратной связи: Запрещено использовать фразы «Мне нравится» или «Это некрасиво». Разрешены только формулировки: «Нарушено правило №2», «Функция X выдает ошибку при значении Y».

Результат: Ученик учится давать конструктивную, деперсонализированную обратную связь. Когда он находит пять ошибок у соседа, он начинает автоматически искать их и в своем коде, прежде чем нажать кнопку «Выполнить».

Общие правила внедрения:

Запрет на стирание: Трассировочная таблица или чек-лист - это летопись. Нельзя зачеркивать ошибки. Нужно провести одну черту и написать правильное число рядом. Это учит видеть историю изменения данных.

Похвала за процесс: При использовании этих приемов хвалите не за правильный ответ, а за факт использования инструмента: «Молодец, ты вовремя поставил красную метку!», «Отлично, ты заметил ошибку в примере соседа быстрее всех!».

3.2.2. Методический анализ выполнения заданий обучающимися с низким уровнем подготовки (в группе от минимального до 60 т.б.), включая методические рекомендации для учителей

3.2.2.1. Описание предметной подготовки участников ЕГЭ с низким уровнем подготовки, анализ типичных дефицитов в подготовке

- Описание достигнутых предметных результатов ФГОС: освоенных тем программы, сформированных умений и т.п. (если имеются)

Участники с низким уровнем подготовки наиболее успешно выполняют задания №1 (61,82%), №2 (73,39%), №3 (57,02%), №4 (74,38%), №10 (81,40%), №12 (44,21%), №15 (14,88%).

Сформированы:

- умение представлять данные в информационных моделях;
- умение строить таблицы истинности;

- навыки поиска в реляционных базах данных;
- информационный поиск средствами текстового процессора;
- навыки кодирования/декодирования информации.

o Анализ типичных дефицитов в предметной подготовке: неосвоенные или слабо освоенные темы программы, несформированные или недостаточно сформированные умения, навыки, виды познавательной деятельности, проявляющиеся в типичных ошибках при выполнении соответствующих заданий

Таблица 2-17

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП
1.	Программирование: создание программ (задания №17, 24–27)	8–11 кл.
2.	Позиционные системы счисления (задание №14)	8, 10 кл.
3.	Основы математической логики (задание №15)	8, 10 кл.
4.	Обработка числовой информации в электронных таблицах (задание №9)	9, 11 кл.
5.	Анализ хода исполнения алгоритма (задание №23)	10–11 кл.
6.	Кодирование и измерение информации (задания №8, 11)	7, 10–11 кл.

Типичные ошибки:

- задание 14 (успешность выполнения 11,69%) – ошибки в нахождении значения выражения в пятеричной системе счисления, при переводе чисел из одной системы счисления в другую; отсутствие ответа.
- задание 15 (успешность выполнения 14,92%) - ошибки при выполнении логических операций, связанные с незнанием основных понятий и законов математической логики; отсутствие ответа.
- задание 9 (успешность выполнения 20,56%) – ошибки при обработке числовой информации в электронных таблицах; обучающиеся учитывали только одно условие из двух; отсутствие ответа.
- задания 24 – 27 (успешность выполнения 0%) – ошибки при создании программ, связанные с неумением составлять программы, незнанием методов обработки символьной информации, целочисленной информации, анализа данных; отсутствие ответа.
- задание 8 (успешность выполнения 35,48%) – ошибки в использовании методов измерения количества информации; отдельные обучающиеся пытались выполнить задание методом подбора; отсутствие ответа.
- задание 11 (успешность выполнения 25%) – ошибки при подсчете информационного объёма сообщения, определении взаимосвязи между числом серийных номеров и объемом памяти в базе данных; вычислительные ошибки; отсутствие ответа.
- задание 17 (успешность выполнения 6,45%) – ошибки при составлении алгоритма обработки числовой последовательности, обучающиеся учитывали только одно условие из двух; отсутствие ответа.

- Реалистичные «зоны роста» в части предметной подготовки для уверенного получения 60-70 т.б.

Для достижения результатов 60-70 т.б. участникам данной группы необходимо:

1. Усилить подготовку по разделу «Алгоритмы и программирование» – освоить написание простых программ (10-15 строк) на языке Python (задание №17).
2. Повысить уровень знаний в области позиционных систем счисления (задание №14) и алгебры логики (№15).
3. Отработать навыки обработки числовой информации в электронных таблицах (задание №9).
4. Систематизировать знания по анализу данных с использованием сортировки (задание №26).
5. Уделить внимание теме «Кодирование информации» и определению объёма памяти (задания №7, №11). Целесообразно организовать индивидуальные траектории подготовки с акцентом на практическое программирование.

3.2.2.2. Методический анализ качества освоения метапредметных результатов ФГОС участниками ЕГЭ с низким уровнем подготовки

- Развернутый комментарий на основе заданий / групп заданий, на успешность выполнения которых могла повлиять достаточная или недостаточная сформированность метапредметных умений, с указанием соответствующих метапредметных умений и их проявления в типичных ошибках в ответах участников экзамена на соответствующие задания;

Допущенные обучающимися данной группы типичные ошибки, рассмотренные выше, позволяют сделать предположение о том, что частично сформированы познавательные УУД, связанные с поиском и обработкой информации.

Однако недостаточно развиты: исследовательские действия (анализ данных, интерпретация результатов); знаково-символические действия при работе с системами счисления; регулятивные действия (планирование последовательности действий при решении многошаговых задач).

Коммуникативные УУД развиты слабо, что проявляется в неумении формализовать условие задачи.

Все задания основаны на сформированных умениях самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях. Для успешного выполнения всех заданий необходимо сформированное умение самостоятельно составлять план решения проблемы с учетом имеющихся ресурсов, собственных возможностей и предпочтений. Значительное число ошибок и трудностей обучающихся при выполнении заданий КИМ может быть обусловлено слабой сформированностью данных умений.

Все задания требуют осуществления пошагового и итогового самоконтроля, умения вносить коррективы в деятельность, оценивать соответствие результатов целям. Недостаточной сформированностью данного умения обусловлено значительное число ошибок и трудностей обучающихся при выполнении заданий КИМ.

- Предположение о достигнутых и недостигнутых метапредметных результатах ФГОС

Метапредметные результаты ФГОС у группы с низким уровнем подготовки достигнуты на близком к базовому уровне. Хорошо сформированы познавательные действия по работе с информацией (поиск, анализ, структурирование). Требуют дальнейшего развития: регулятивные действия (самоконтроль, коррекция), исследовательские действия (анализ данных, выявление закономерностей). Рекомендуется включать в уроки задания на планирование и прогнозирование результатов.

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

Общие рекомендации:

1. Использовать метод «Пошаговой декомпозиции» при решении алгоритмических задач – разбивать сложные задачи на подзадачи.
2. Практиковать задания на разработку небольших программ (10-15 строк) с обязательной проверкой на разных входных данных.
3. Проводить разбор типовых ошибок через анализ некорректно работающего кода.
4. Уделять больше внимания теме «Системы счисления», в том числе с использованием интерактивных тренажёров.
5. Регулярно проводить алгоритмические разминки – короткие задания на анализ готовых алгоритмов.
6. Использовать практико-ориентированные задания (расчёт бюджета, планирование маршрута).
7. Организовать взаимную проверку программ в парах.

Приведем конкретные задания и методические приемы по сложным для обучающихся аспектам.

Обучение слабых учеников программированию требует отказа от классического академического подхода («сначала теория типов данных, потом синтаксис»). Для них код – это не текст, а последовательность команд для «глупого робота».

Главная цель на этапе 10–15 строк: избежать страха ошибки и научить видеть структуру программы.

Вот пошаговая методика организации работы.

Этап 1. Подготовка среды (Снятие технических барьеров)

Слабый ученик часто ломается еще до написания первой строки из-за опечаток в английской раскладке или ошибок установки ПО.

Инструмент: Используйте онлайн-компиляторы (например, Replit) или режимы с подсветкой синтаксиса, где нельзя запустить код, если скобка не закрыта.

Правило шаблона: Не заставляйте писать программу с нуля. Дайте файл-заготовку (template.py), где уже написаны служебные слова:

python Копировать

```python
# Мое имя: _______
print("Программа запущена")

# Твой код здесь

print("Конец работы")
input("Нажми Enter для выхода") # Чтобы окно не закрывалось само
```

Ученик заполняет только середину. Это дает чувство завершенности.

Этап 2. Визуализация алгоритма до кода (Блок-схемы-раскраски)

Прежде чем написать x = int(input()), ученик должен нарисовать ромб или прямоугольник.

Метод «Лего»: Выдайте листы с напечатанными пустыми блоками (начало, ввод, процесс, вывод, конец). Ученик сначала составляет схему задачи карандашом.

Цветовое кодирование:

Синий - то, что компьютер спрашивает у человека (input).

Красный - то, что компьютер считает сам (+, -, *).

Зеленый - то, что компьютер говорит человеку (print).

Правило: Код пишется строго по стрелочкам схемы. Нет схемы - нет доступа к компьютеру.

Этап 3. Магия присваивания (Самый сложный момент)

Строка a = a + 1 вызывает у слабых учеников когнитивный диссонанс (математически это неверно).

Аналогия «Мусорный бак и коробка»:

Слева от знака = стоит коробка. Мы можем приклеить на неё бумажку с именем a.

Справа от знака = лежит мусор, который мы сейчас подберем.

Знак = — это действие: «Выброси старый мусор из коробки a и положи туда новый мусор справа».

Упражнение «Робот-руки»: Два ученика. Один - «Переменная», второй - «Оператор». Оператор берет со стола карточку с числом и физически вкладывает её в руки «Переменной». Если нужно сделать a = a + 5, оператор смотрит на число в руках напарника, прибавляет 5, берет новую карточку и меняет старую на новую.

Этап 4. Жесткая структура кода (Паттерн IPO: Input – Process – Output)

Любая программа на 10–15 строк должна писаться только в этом порядке. Нарушение порядка карается немедленным удалением кода.

INPUT (Ввод): Сначала собираем все данные.

Все переменные должны быть объявлены в самом верху.

Сразу делайте проверку: age = int(input(«Введите возраст: «)). Учите писать подсказки внутри input, чтобы понимать, какую цифру просит машина.

PROCESS (Процесс): Только вычисления.

Никаких выводов (print) в середине расчетов.

Каждая математическая операция — на новой строке. Не пишите s = (a+b)*c/2. Разбейте:

```python
sum_ab = a + b
prod = sum_ab * c
s = prod / 2
```

OUTPUT (Вывод): Красивый ответ в конце.

Использование f-строк сразу: print(f»Площадь равна {s}»).

Этап 5. Отладка без паники (Трассировка руками)

Когда программа выдает ошибку или неверный результат, слабый ученик обычно просто переписывает её заново или зовет учителя.

«Дневник Робота»: Дайте таблицу трассировки. Ученик обязан заполнить её вручную перед тем, как менять код.

Шаг	Переменная x	Переменная y	Что делает эта строчка?
1	5	?	x = 5
2	5	10	y = x * 2
3	3	10	x = x - 2

Правило «Закрытой крышки»: При вычислении значения правой части закройте пальцем всё остальное. Берите значения только сверху.

Примеры цепочки заданий (От простого к сложному)

Не давайте разрозненные задачи. Давайте развитие одной идеи.

1. Приветствие: Напиши код, который спросит имя и скажет «Привет, [Имя]!». (Только input и print).
2. Калькулятор одного действия: Спроси два числа и выведи их сумму.
3. Магазин: Спроси цену товара и количество. Выведи общую стоимость. (Вводим умножение).
4. Сдача: Узнай у пользователя, сколько денег он дал продавцу, и посчитай сдачу. (Вводим вычитание).

5. Таймер варки яйца: Известно, что яйцо варится 7 минут. Узнай текущее время (в минутах от начала урока) и скажи, во сколько его выключать. (Вводим сложение времени).

6. Удвоитель (Цикл for): Программа 5 раз спрашивает число и каждый раз пишет его удвоенное значение. (Здесь важно показать копипасту и объяснить, почему цикл лучше).

Чек-лист для самопроверки учеником (Наклейте на монитор):

- Есть ли у меня схема на бумаге?
- Собрал ли я ВСЕ данные через input в начале?
- Использовал ли я английскую раскладку для имен переменных?
- Поставил ли я двоеточие после условия/цикла?
- Отступил ли я код внутри цикла пробелами (Tab)?
- Проверил ли я расчет на черновике (трассировка)?

Для отработки навыков работы с числовыми данными в электронных таблицах (Excel) у учеников важно уйти от абстрактных «посчитайте сумму» к решению понятных жизненных проблем. Слабые ученики часто путают относительные и абсолютные ссылки, а также не понимают разницы между форматом ячейки и её реальным значением.

Вот три методических приема для формирования устойчивых навыков обработки чисел.

1. Прием «Магазин без калькулятора» (Формулы и адресация)

Цель: Научить использовать математические функции (СУММ, СРЗНАЧ) и закрепить разницу между относительной (A1) и абсолютной (B1) ссылкой через реальный быт.

Сценарий: Ученик - владелец маленького магазина. У него есть список товаров, закупочная цена и наценка. Наценка одинаковая для всех (например, 20%).

Алгоритм действий:

1. Подготовка данных: Столбец A - Товар, Столбец B - Закупка.
2. Ловушка (Относительная ссылка): В ячейке C2 ученик пишет формулу наценки: =B2*20%. Он протягивает её вниз. Всё работает.
3. Критическая точка (Абсолютная ссылка): Учитель говорит: «Налоги выросли, теперь наценка должна быть 25% для всех. Измени только одну ячейку».

Если ученик впишет 25% в каждую строку - он проиграл.

Правильный путь: создать отдельную ячейку (например, F1) со значением 0.25.

Формула меняется на =B2*F1.

4. Проверка навыка: Теперь при изменении цифры в F1 пересчитывается весь столбец автоматически.

5. Дополнительный слой: Добавьте столбец «Количество». Задача: посчитать общую выручку (Цена продажи * Количество), используя автосумму (∑).

Почему это работает: Ученик видит физический результат изменения одной цифры («О! Все цены сами поменялись!»). Это формирует понимание того, что таблица - это динамическая модель, а не просто бумага.

2. Прием «Страж логики» (Логические функции и фильтрация)

Цель: Освоение функций ЕСЛИ (IF), И (AND), ИЛИ (OR) и расширенного фильтра. Переход от простых вычислений к принятию решений.

Сценарий: Обработка результатов школьного забега или тестирования. Нужно выделить группы учеников по критериям.

Задание: Дана таблица: Фамилия, Класс, Баллы за тест (0–100). Нужно автоматизировать выставление оценок и определение статуса:

1. Простое условие: В столбце «Оценка» пишем =ЕСЛИ(C2>=85; «Отлично»; «Норма»).

2. Сложное условие (Квалификация): Создаем колонку «Призер».

Условие: Призером считается тот, кто из 10 класса И набрал больше 90 баллов.

Формула: =ЕСЛИ(И(B2=10; C2>90); «Да»; «Нет»).

3. Визуальный контроль (Условное форматирование):

Попросите учеников покрасить всех «Призеров» в зеленый цвет, а тех, кто набрал меньше 50 баллов - в красный.

Важно: Научите их делать это не вручную цветом заливки, а через инструмент «Условное форматирование», чтобы правило сохранялось при добавлении новых строк.

4. Финал (Фильтр): Используйте автофильтр, чтобы показать только тех, кому нужно «Пересдача».

Почему это работает: Функция ЕСЛИ сложна своей вложенностью. Разбиение задачи на визуальные критерии (цвета) помогает слабым ученикам сначала увидеть логику глазами, а потом написать её формулой.

3. Прием «Детектив данных» (Агрегированные вычисления и защита от ошибок)

Цель: Отработка статистических функций (МИН, МАКС, СЧЁТЕСЛИ, СУММЕСЛИ) и поиск скрытых аномалий в данных.

Сценарий: Анализ успеваемости по параллели или продаж вымышленного интернет-магазина. Данные содержат «мусор».

Этапы расследования:

Поиск экстремумов: Найти самого успешного/самого неуспешного. Функции МАКС и МИН.

Подсчет количества (СУММЕСЛИ / СЧЁТЕСЛИ):

Вопрос: «Сколько человек получили двойку?».

Вместо ручного выделения ячеек используем =СЧЁТЕСЛИ(B2:B50; «<=34»).

Вопрос: «На какую сумму продали только товар «Хлеб»?».

Используем =СУММЕСЛИ(A2:A100; «Хлеб»; C2:C100).

Борьба с типографикой (Навык очистки):

Учитель дает файл, где числа записаны через запятую вместо точки, или где перед числом стоит лишний пробел. Сумма не считается.

Прием решения: Использование функции поиска и замены (Ctrl+H) и функции ЗНАЧЕН (VALUE), которая превращает текстовый вид числа в настоящее число.

Округление: Расчет средней цены товара. Объясните, почему нельзя оставлять 15 знаков после запятой (ОКРУГЛ).

Почему это работает: Задание превращается в квест. Ученик чувствует себя аналитиком, который нашел ошибку в чужих расчетах. Очистка данных — критически важный навык, которому редко учат специально, но который постоянно встречается в реальности.

Общие советы для учителя:

- Запрет на ручной пересчет: Если задачу можно решить, потянув формулу за уголок — она должна быть решена именно так. Любое переписывание цифр руками = минус балл.
- Формат vs Значение: Регулярно задавайте вопрос: «Что увидит компьютер, если мы попросим его найти среднее от этой даты?». Это учит понимать, как Excel хранит данные внутри.

Для слабых учеников главная трудность в теме «Кодирование» - это абстрактность формул и путаница между количеством символов (сколько букв написано) и весом одного символа (сколько места занимает одна буква). Также пугает необходимость возводить двойку в степень.

Задачи должны опираться на визуальные образы: коробки, ячейки памяти и простые пропорции.

Вот три задания на определение информационного объема для слабых учеников, где акцент сделан на визуальные сетки, таблицы поиска и бытовые ограничения памяти.

Задание 1. «Морской бой: Кодирование координат»

Цель: Научить определять информационный объем через степень двойки ($2^i \geq N$), используя игровую сетку вместо абстрактного алфавита. Проблема ученика: Страх перед формулой $i = \log_2 N$.

Условие: Мы играем в расширенный морской бой на огромном поле размером 15 на 15 клеток. Нам нужно передать координату одной клетки по рации.

Какое минимальное количество бит потребуется, чтобы закодировать номер строки?

Сколько бит нужно, чтобы закодировать всю клетку целиком (и букву/цифру)?

Пошаговый алгоритм («Лестница степеней»): Слабым ученикам нельзя давать логарифм. Дайте им таблицу-помощник:

Степень	Значение (2^i)
1	2
2	4
3	8
4	16

5 32

Решение:

У нас есть 15 строк. Смотрим в таблицу: число 15 не входит в 8 (это мало), но оно меньше 16.

Значит, для номера строки нужно 4 бита (так как 4-я ступень дает нам 16 вариантов - этого хватит с запасом).

Для столбца всё то же самое: тоже 4 бита.

Чтобы передать всю точку (адрес), мы должны соединить эти данные: 4 + 4 = 8 бит (ровно 1 байт).

Почему это работает: Мы заменяем страшное слово «алфавит» понятной сеткой игры. Ученик видит, что если бы поле было 16x16, ответ был бы ровно 8 бит, а так как у нас 15x15 - нам все равно нужно столько же места, сколько для 16-ти.

Задание 2. «Пароль-супергерой» (Комбинированный объем)

Цель: Отработать формулу V = K * i в ситуации, когда символы имеют разный вес (буквы vs цифры). Понять разницу между мощностью смешанного алфавита и объемом конкретной строки.

Условие: Система безопасности требует создавать пароли длиной ровно 6 символов. Правила такие:

Первые 2 символа - это заглавные русские буквы (в алфавите 33 буквы).

Последние 4 символа - это цифры (от 0 до 9).

Сколько информации несет один такой пароль? Ответ дайте в байтах.

Метод «Разных корзин»: Не складываем мощности алфавитов! Сначала считаем информацию по частям.

Часть А (Буквы):

Букв 33. Смотрим в «Лестницу степеней» из прошлого задания. 32 — мало, значит берем следующую ступень — 6 бит на одну букву.

У нас две буквы: 6 + 6 = 12 бит.

Часть Б (Цифры):

Цифр 10. В таблице: 8 - мало, значит берем 4 бита на одну цифру (ступень №4).

У нас четыре цифры: 4 * 4 = 16 бит.

Итого за весь пароль:

Складываем части: 12 + 16 = 28 бит.

Перевод в байты (Главный шаг):

Напоминаем правило: «В одном байте живут 8 бит».

Делим: 28 / 8 = 3,5 байта.

Типичная ошибка: Ученики пытаются сложить 33 + 10 = 43 знака и считать $\log_2 43$. Метод «Разных корзин» физически разделяет типы данных в голове ребенка.

Задание 3. «Очистка памяти смартфона» (Вычисление остатка)

Цель: Показать, что память устройства ограничена системными файлами, и научить вычитать объемы разных типов файлов.

Контекст: Управление хранилищем без использования сторонних приложений.
Условие: У вас есть флешка объемом 1 Гигабайт (ГБ). Вы знаете, что:
Система Android уже заняла 200 Мегабайт (МБ) под свои нужды.
Одно ваше фото занимает 4 МБ.
Один видеоролик занимает 50 МБ.
Вопросы:
Сколько фотографий поместится на флешку, если видео записывать не будем?
Если мы запишем одно видео, сколько фото поместится на оставшееся место?
Алгоритм «Свободная полка»: Главная проблема здесь - переход от ГБ к МБ.
Приведение к общему виду:
Правило: «Чтобы получить Мегабайты из Гигабайт, умножь примерно на 1000» (для слабых учеников используем упрощенное 10^3, чтобы не грузить 1024).
Флешка: 1 ГБ = 1000 МБ.
Шаг 1: Вычет системы:
1000 - 200 = 800 МБ - это наша «свободная полка».
Шаг 2: Только фото:
800 / 4 = 200 фотографий.
Шаг 3: Фото после видео (Критический контроль):
Сначала тратим деньги на дорогое видео: 800 - 50 = 750 МБ осталось.
Теперь делим остаток на фото: 750 / 4 = 187,5.
Важный момент: Половинку фото записать нельзя. Округляем в меньшую сторону до целого.
Ответ: 187 фото.

Почему это полезно: Это учит многоступенчатому расчету и внимательности к условиям (сначала вычесть одно, потом разделить другое). Также это готовит их к реальности, где память телефона всегда «меньше заявленной».

3.2.3. Методический анализ выполнения заданий обучающимися со средним уровнем подготовки (в группе от 61 до 80 т.б.), включая методические рекомендации для учителей

3.2.3.1. Описание предметной подготовки участников ЕГЭ со средним уровнем подготовки, анализ типичных дефицитов в подготовке

o Описание достигнутых предметных результатов ФГОС: освоенных тем программы, сформированных умений и т.п. (если имеются)

Участники со средним уровнем подготовки демонстрируют уверенное выполнение большинства заданий базового и повышенного уровня.

Наиболее сформированы:

- умение представлять данные в разных типах информационных моделей (№1 – 93,55%);
- умение строить таблицы истинности (№2 – 93,52%);
- навыки поиска в базах данных (№3 – 80,16%);
- информационный поиск (№10 – 88,26%);
- навыки формального исполнения алгоритмов (№5 – 86,64%, №6 – 83,40%);
- вычисление рекуррентных выражений (№16 – 88,52%).

Также хорошо освоены:

- анализ логической игры (№19 – 82,19%),
- поиск выигрышной стратегии (№20 – 85,83%),
- анализ хода исполнения алгоритма (№23 – 48,58%).

o Анализ типичных дефицитов в предметной подготовке: неосвоенные или слабо освоенные темы программы, несформированные или недостаточно сформированные умения, навыки, виды познавательной деятельности, проявляющиеся в типичных ошибках при выполнении соответствующих заданий

Таблица 2-18

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП
1.	Программирование: обработка символьной информации (задание №24)	10–11 кл.
2.	Программирование: обработка целочисленной информации (задание №25)	10–11 кл.
3.	Сортировка и обработка массивов (задание №26)	11 кл.
4.	Анализ данных (задание №27)	11 кл.
5.	Работа с электронными таблицами (задание №18)	9, 11 кл.

6.	Использование маски подсети (задание №13)	10 кл.

Типичные ошибки:

- задание 24 (успешность выполнения 0,81) - ошибки при создании программ, связанные с неверным применением методов обработки символьной информации; отсутствие ответа.
- задание 25 (успешность выполнения 7,29%) - ошибки при создании программ, связанные с неверным применением методов обработки целочисленной информации; отсутствие ответа.
- задание 26 (успешность выполнения 1,62%) – ошибки при применении методов сортировки и обработки массивов; в ответе требуется записать два числа: сначала идентификатор устройства, с которого на сервер был передан наибольший суммарный объём данных не позднее 11:59:59, а затем сумму объёмов двух наибольших резервных копий специального раздела (в Кбайт); обучающиеся указали только одно число, частично выполнив условие.
- задание 27 (успешность выполнения 3,44%) – ошибки связаны с невнимательным прочтением условия задания; в ответе требовалось записать четыре числа: в первой строке – сначала A1, затем целую часть произведения A2 × 10 000; во второй строке – сначала целую часть произведения B1 × 10 000, затем B2; обучающиеся записали ответ в другой последовательности или записали менее четырех чисел.

- Реалистичные «зоны роста» в части предметной подготовки для уверенного получения 80 и более т.б.

Для достижения результатов 80+ т.б. участникам данной группы необходимо:

1. Освоить программирование обработки символьной информации (задание №24) – научиться создавать программы 10-20 строк для работы со строками.
2. Углубить навыки программирования обработки целочисленной информации (задание №25).
3. Повысить уровень владения сортировкой и обработкой больших массивов данных (задание №26).
4. Освоить последовательность решения задач анализа данных (задание №27).
5. Улучшить навыки работы с электронными таблицами для обработки целочисленных данных (№18).
6. Закрепить знания по маскированию IP-сетей (№13) и кодированию информации (№4). Целесообразно активное использование сред программирования (Python, C++) при решении задач.

3.2.3.2. Методический анализ качества освоения метапредметных результатов ФГОС участниками ЕГЭ со средним уровнем подготовки

- Развернутый комментарий на основе заданий / групп заданий, на успешность выполнения которых могла повлиять достаточная или недостаточная сформированность метапредметных умений, с указанием соответствующих метапредметных умений и их проявления в типичных ошибках в ответах участников экзамена на соответствующие задания;

У участников со средним уровнем подготовки достаточно хорошо сформированы познавательные УУД: анализ, синтез, сравнение, обобщение. Развиты знаково-символические действия (умение читать схемы, таблицы, графы).

Наличие типичных ошибок, рассмотренных выше, свидетельствует от том, что недостаточно сформированы: исследовательские действия (анализ данных, интерпретация), регулятивные действия при решении комплексных задач (планирование многошагового решения, самоконтроль на каждом этапе). Коммуникативные УУД проявляются в умении формализовать условие задачи.

- Предположение о достигнутых и недостигнутых метапредметных результатах ФГОС

Метапредметные результаты ФГОС у группы со средним уровнем подготовки достигнуты в достаточной степени. Хорошо сформированы познавательные и знаково-символические УУД. Рекомендуется уделить внимание развитию исследовательских действий (анализ данных, построение моделей) и регулятивных действий (самоконтроль, коррекция, оценка результатов).

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

Общие рекомендации:

1. Активно использовать среды программирования (Python, C++, Pascal) для решения задач обработки символьной и целочисленной информации (задания №24, №25).
2. Применять метод «Обратного проектирования» – восстановление исходных данных по результатам работы программы.
3. Практиковать решение задач анализа данных (задание №27) с использованием электронных таблиц и сред программирования.
4. Разбирать типовые ошибки при работе с массивами и сортировкой (задание №26).
5. Включать в учебный процесс мини-проекты (создание простых игр, обработка данных) с презентацией готового продукта.
6. Использовать чек-листы для самопроверки программ.
7. Организовать работу в парах с взаимной проверкой и обсуждением решений.

Приведем конкретные приемы и примеры заданий.

Чтобы углубить навыки программирования в области обработки целых чисел, необходимо перейти от простых арифметических действий к теории чисел, двоичной логике и эффективным алгоритмам. Для учеников 9–11 классов это мост между школьной информатикой и олимпиадным программированием.

Работа должна строиться по принципу «от инструмента к задаче»: сначала осваивается конкретная техника (например, работа с битами), а затем она применяется для решения нетривиальных проблем.

Блок 1. Фундамент: Арифметика остатков и делимость

Многие задачи на числа решаются без циклов деления, если знать свойства остатков.

Темы для изучения:

Нахождение последней цифры (n (mod 10)) и первой цифры (циклическое деление на 10).

Признаки делимости через остатки (например, число делится на 3, если сумма его цифр делится на 3).

Алгоритм Евклида для поиска НОД (Наибольшего общего делителя) - классический пример рекурсии или цикла.

Задание-практикум: «Криптографическая контрольная сумма»

Написать функцию checksum(number), которая вычисляет сумму последних цифр всех простых множителей числа.

Пример: Число 60 = 2 * 2 * 3 * 5. Простые множители: 2, 3, 5. Сумма последних цифр: 2 + 3 + 5 = 10. Сложность: Ученику придется написать эффективный цикл проверки на простоту и разложения на множители, избегая перебора до самого числа N.

Блок 2. Продвинутый уровень: Двоичное представление (Битовые операции)

Целые числа в памяти компьютера - это наборы битов. Работа с ними происходит мгновенно и часто заменяет сложные условия.

Инструментарий: Операции & (И), | (ИЛИ), ^ (XOR), << (сдвиг влево), >> (сдвиг вправо).

Темы для изучения:

Четность: x & 1 == 0.

Умножение/деление на степени двойки через сдвиги (x << 1 это x * 2).

Изоляция конкретного бита.

Быстрый обмен значениями без третьей переменной: a ^= b; b ^= a; a ^= b;.

Задание-практикум: «Анализ регистров датчика» Датчик передает состояние в виде одного целого числа (маски). Например, число 13 в двоичном виде - 1101. Ученику нужно написать программу-анализатор:

Вывести состояние устройства в виде списка флагов («Ошибка», «Готовность», «Питание»).

Реализовать включение/выключение конкретного флага (например, включить «Звук», не меняя остальные биты). Математический аспект: Здесь используется формула NewValue = OldValue | MASK (для включения) и NewValue = OldValue & ~MASK (для выключения).

Блок 3. Генерация и перебор: Целочисленные последовательности

Задачи, где нужно найти все комбинации или проверить диапазон чисел на соответствие свойствам.

Темы для изучения:

Перебор цифр числа (через % 10 и // 10) vs перевод в строку. Важно показать разницу в скорости и типах данных.

Поиск палиндромов среди чисел.

«Красивые числа»: числа, у которых сумма цифр равна их произведению.

Генерация пифагоровых троек (a^2 + b^2 = c^2) эффективным способом.

Задание-практикум: «Числа Армстронга (Самовлюбленные числа)» Найти все трехзначные числа, которые равны сумме кубов своих цифр. Пример: 153 = 1^3 + 5^3 + 3^3.

Углубление: Предложите ученикам оптимизировать поиск. Вместо тройного вложенного цикла (for i, for j, for k) можно использовать один цикл от 100 до 999 и одну функцию извлечения цифр. Затем расширить задачу до 4-х и 5-значных чисел.

Блок 4. Эффективность: Ограниченные диапазоны (Решето Эратосфена)

Когда нужно обработать много чисел сразу (например, найти все простые числа до миллиона), обычные циклы работают слишком медленно.

Метод: Использование списков булевых значений (массивов флагов).

Задание-практикум: «Генератор простых чисел для RSA»

Реализовать алгоритм «Решето Эратосфена». Создать массив is_prime[1...1000000].

За доли секунды отметить составные числа.

Вывести количество простых чисел в диапазоне [L, R], которое пользователь вводит вручную. Навыки: Это учит работать с памятью (списками) и понимать сложность алгоритмов.

Методические рекомендации по организации работы:

- Визуализация памяти: Используйте онлайн-инструменты (или просто рисуйте на доске ячейки памяти), чтобы показать, как число 5 превращается в 101. Слабым местом всегда остается непонимание разницы между числом три и кодом один-один.
- Ограничение встроенных функций: Запретите использование bin(), str() и математических библиотек при изучении базовых техник. Ученик должен сам написать функцию перевода в другую систему счисления через остаток от деления.
- Бенчмаркинг (Сравнение скоростей): Давайте одну и ту же задачу (например, поиск простых чисел) решить двумя способами: «в лоб» (перебор) и эффективно (решето/корень). Покажите время выполнения. Для сильных детей это лучший мотиватор писать оптимальный код.
- Разбор чужого кода: Приносите на разбор короткие (5–10 строк), но хитрые фрагменты кода из олимпиад (например, быстрый обратный элемент или подсчет единичных битов - алгоритм Кернигана). Попросите класс предсказать результат выполнения без запуска.

Обработка символьной информации (строк) - это переход от работы с числами к работе с текстом. Для учеников главная сложность здесь заключается в понимании того, что строка - это массив символов, где у каждого есть свой индекс (адрес), и что регистр букв ('A' и 'a') для компьютера - это разные сущности.

Вот три задания разного типа: детективный поиск, лингвистическая статистика и криптография.

Задание 1. «Детективное агентство: Поиск цифровых улик» (Работа с методами поиска)

Цель: Научиться находить подстроки, определять их границы и работать с индексами без использования сложных регулярных выражений. Контекст: Анализ логов или перехваченных сообщений.

Условие: Дана длинная строка S (например, системный лог). В ней спрятаны IP-адреса серверов. Известно, что все они начинаются на префикс 192.168.. Нужно найти не сами адреса, а позиции (индексы), где эти подозрительные участки встречаются в тексте.

Алгоритм действий:

Использовать метод .find() (или аналог).

Важно объяснить ученику разницу:

Если мы просто ищем «192», программа найдет его даже внутри числа 51923.

Нам нужно искать именно связку «192.168.».

Цикл поиска: Показать типичную ошибку - бесконечный цикл. Чтобы его избежать, после нахождения индекса pos, следующий поиск должен начинаться с позиции pos + 1.

Результат: Вывести список номеров позиций. Например: Найдено по индексам: 15, 42, 105.

Усложнение («Ловушка регистра»): Дать строку, где написано User logged in from 192.168.... Программа должна найти адрес, несмотря на то, что слово «from» может быть написано как «FROM» или «From». Это учит использовать приведение к единому регистру (.lower() или .upper()) перед поиском.

Задание 2. «Статистический анализ текста» (Частотность и очистка данных)

Цель: Научиться фильтровать символы, удалять лишнее и подсчитывать частоту элементов. Понять концепцию алфавита сообщения. Контекст: Подготовка данных для искусственного интеллекта или расшифровка древних манускриптов.

Условие: Пользователь вводит произвольный текст (стихотворение, предложение из чата). Программа должна выдать отчет:

Сколько всего знаков препинания (точек, запятых, восклицательных знаков)?

Какой процент букв в тексте составляют гласные?

Какая согласная буква встречается чаще всего? (Игнорируя регистр: 'Т' и 'т' считаем за одну букву).

Пошаговый алгоритм («Очистка зерна от плевел»):

Нормализация: Сразу переводим всю строку в нижний регистр: text = input().lower(). Это избавляет от проверок вида if ch == 'a' or ch == 'A'.

Создание словарей-масок:

Создать строку-гласных: vowels = «ауоыиэяюёе».

Создать строку-согласных: consonants = «бвгджзйклмнпрстфхцчшщ».

Проход циклом:

Для каждой буквы ch в строке проверяем:

if ch in vowels: увеличиваем счетчик гласных.

if ch in consonants: увеличиваем значение в словаре частот для этой конкретной буквы.

Финал: Находим максимум в словаре согласных.

Почему это сложно: Ученики часто пытаются сравнивать символ с массивом через ==, вместо использования оператора принадлежности in. Также важно научить их игнорировать пробелы при подсчете статистики.

Задание 3. «Шифр Цезаря 2.0: Сдвиг по кодовой таблице» (ASCII/Unicode арифметика)

Цель: Понять, что символы - это числа, и научиться выполнять над ними математические операции. Освоить циклический сдвиг (переход от 'Я' снова к 'А'). Контекст: Простейшее шифрование сообщений.

Условие: Написать программу-шифратор.

Пользователь вводит текст только заглавными русскими буквами (без пробелов).

Пользователь вводит ключ сдвига K (число).

Каждая буква должна переместиться вправо по алфавиту на K позиций. Буква 'Я', сдвинутая на 1, становится 'А'.

Математическая модель: Компьютер не знает порядка букв, он знает их коды (Юникод). Для русского алфавита (33 буквы):

Получаем код буквы: code = ord(ch).

Приводим к диапазону от 0 до 32: index = code - ord('А').

Делаем сдвиг с учетом кругового обхода (остаток от деления): new_index = (index + K) % 33.

Возвращаем обратно в код буквы: new_code = new_index + ord('А').

Превращаем обратно в символ: new_ch = chr(new_code).

Типичная ошибка-ловушка: Если ученик забудет про оператор % (остаток от деления), то при сдвиге буквы 'Э' на 5 позиций программа выдаст символ спецсимволов или вопросительный знак, так как код выйдет за пределы таблицы русских букв. Использование остатка гарантирует, что после 'Я' (индекс 32) прибавление единицы даст (32+1) % 33 = 0, что соответствует 'А'.

Усложнение: Реализовать дешифрацию. Здесь придется использовать вычитание и следить, чтобы остаток не стал отрицательным (формула (index - K) % 33 работает корректно в Python, но требует пояснения в других языках).

3.2.4. Методический анализ выполнения заданий обучающимися с высоким уровнем подготовки (в группе от 81 т.б. и выше), включая методические рекомендации для учителей

3.2.4.1. Описание предметной подготовки участников ЕГЭ с высоким уровнем подготовки, анализ типичных дефицитов в подготовке

- Описание достигнутых предметных результатов ФГОС: освоенных тем программы, сформированных умений и т.п. (если имеются)

Участники с высоким уровнем подготовки демонстрируют уверенное выполнение практически всех заданий базового и повышенного уровня.

Наиболее успешно выполняются задания №1 (99,13%), №2 (96,52%), №3 (93,04%), №4 (97,39%), №5 (99,13%), №6 (92,17%), №7 (99,13%), №8 (86,96%), №9 (92,17%), №10 (97,39%).

Сформированы навыки программирования: задание №17 (95,65%), задание №24 (40,87%), задание №25 (59,13%), задание №26 (34,35% – 1 балл, 25,69% – 2 балла), задание №27 (58,70% – 1 или 2 балла).

Участники данной группы владеют полным спектром предметных знаний, предусмотренных ФГОС СОО на углублённом уровне.

- Анализ типичных дефицитов в предметной подготовке: неосвоенные или слабо освоенные темы программы, несформированные или недостаточно сформированные умения, навыки, виды познавательной деятельности, проявляющиеся в типичных ошибках при выполнении соответствующих заданий

Таблица 2-19

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП
1.	Программирование: оптимальные алгоритмы обработки символьной информации	10–11 кл.
2.	Программирование: эффективные алгоритмы обработки целочисленной информации	10–11 кл.
3.	Сортировка и обработка больших массивов данных (задание №26)	11 кл.
4.	Комплексный анализ данных: сбор, очистка, моделирование, интерпретация (№27)	11 кл.
5.	Теоретические основы информатики: системы счисления, кодирование, логика	7–11 кл.

- Реалистичные «зоны роста» в части предметной подготовки для получения 100 т.б.

Для получения 100 т.б. участникам с высоким уровнем подготовки необходимо:

1. Достичь стабильного выполнения заданий №26 и №27 на максимальный балл – отработать навыки сортировки больших массивов данных и анализа данных.
2. Повысить эффективность решения задач на обработку символьной информации (№24) – научиться находить оптимальные алгоритмы.
3. Довести до автоматизма написание программ для обработки целочисленной информации (№25).
4. Углубить знания в области теоретических основ информатики для безошибочного выполнения заданий №8, №11, №13, №14.

5. Развивать навыки тестирования и отладки программ, работы с нестандартными входными данными.
6. Освоить методы оптимизации алгоритмов для обработки больших данных.

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

Общие рекомендации:

1. Предоставить участникам с высоким уровнем подготовки задания повышенной сложности, в том числе из олимпиадных задач по программированию.
2. Организовать участие в хакатонах, проектной деятельности, олимпиадах по информатике.
3. Практиковать написание эффективных алгоритмов с оценкой временной сложности.
4. Использовать платформу «Сириус Онлайн», проект «Код Будущего» для дополнительного образования.
5. Рекомендовать самостоятельное изучение продвинутых структур данных и алгоритмов (динамическое программирование, теория графов).
6. Проводить тренировочные экзамены в формате ЕГЭ с полным разбором результатов.

Для сильных учеников стандартные задачи на вычисление или реализацию известных алгоритмов не представляют интереса. Им необходимы задания, где нет готового ответа в конце учебника, требуется самостоятельный поиск информации и синтез знаний из разных областей.

Вот пять примеров заданий творческого, поискового и межпредметного характера для 8–11 классов.

1. Творческое задание: «Генеративная поэзия Маркова» (Строки + Теория вероятностей)

Суть: Создание программы, которая анализирует творчество выбранного поэта и генерирует новые стихи, сохраняя его стиль.

Задача: Написать программу, которая принимает на вход текстовый файл со стихами (например, Серебряный век) и строит модель цепей Маркова.

Механика: Программа должна составить словарь пар слов («текущее слово» → «список возможных следующих слов»). При генерации программа выбирает следующее слово случайно, но с учетом вероятности появления именно после предыдущего.

Межпредметность: Информатика + Литература. Ученик должен сам выбрать автора, найти корпус текстов (возможно, очистить их от лишних символов), а затем оценить результат - насколько «машинный» стих похож на оригинал стилистически.

Усложнение: Реализовать сохранение рифмы (поиск слов, оканчивающихся на одинаковые сочетания букв) и стихотворного размера (подсчет слогов).

2. Поисковое задание: «Цифровая археология утраченных форматов»

Суть: Восстановление данных из устаревших или нестандартных структур без документации.

Задача: Учитель предоставляет бинарный (не текстовый) файл небольшого объема, имитирующий старый формат сохранения игры или проприетарный графический формат. Документации к нему нет.

Инструментарий: Hex-редактор (просмотр байтов).

Ход работы:

Найти «магические числа» (сигнатуры файлов, например 42 4D для BMP).

Определить размер заголовка.

Вычислить смещение, где начинаются данные о пикселях или координатах объектов.

Написать Python-скрипт, который читает этот файл как поток байтов (rb), извлекает нужные значения через побитовые операции и визуализирует результат (рисует картинку черепашкой или выводит карту уровня).

Поисковая часть: Ученику придется гуглить спецификации старых форматов (например, палитры VGA или структуру карт Doom), чтобы понять логику упаковки цветов.

3. Межпредметное задание: «Биоинформатика: Охотник за генами» (Обработка строк + Биология)

Суть: Применение классических алгоритмов информатики для анализа ДНК.

Задача: Анализ последовательности нуклеотидов (строка из букв A, T, G, C).

Алгоритмическая часть:

Написать эффективный алгоритм поиска паттерна (алгоритм Кнута-Морриса-Пратта или Бойера-Мора). Простой перебор (O(N* M)) будет работать слишком долго на реальных геномах.

Найти наиболее часто встречающийся фрагмент длиной K (k-меры).

Посчитать показатель CG-наполненности (важен для определения стабильности генома).

Биологический контекст: Ученик должен исследовать, что такое промоторы, кодирующие участки (экзоны) и интроны. Задача усложняется: нужно найти открытые рамки считывания (ORF) - участки, начинающиеся со старт-кодона ATG и заканчивающиеся стоп-кодоном.

4. Творческое задание: «Кибернетическая музыка» (Функции + MIDI/Аудио)

Суть: Синтез звука математическими функциями вместо использования готовых семплов.

Задача: Создать мелодию, используя ряды Фурье или базовые волны (синус, пила, квадрат).

Реализация:

Использовать библиотеки типа numpy для генерации массивов значений амплитуды и scipy.io.wavfile для записи в .wav.

Математическая модель: Звук ноты «Ля» - это синусоида частотой 440 Гц. Аккорд - это сумма нескольких синусоид.

Ритм реализуется через огибающую (изменение громкости во времени).

Творческий аспект: Ученик выступает как саунд-дизайнер. Он может попытаться воссоздать звук сирены, барабана или космического лазера, комбинируя формулы и управляя параметрами сигнала (частота, фаза, затухание).

5. Поисковое задание: «Визуализация открытых данных» (Парсинг + Статистика)

Суть: Самостоятельный сбор данных из интернета и создание интерактивной аналитической модели.

Задача: Выбрать тему (например, «Цены на видеокарты RTX за последний год», «Статистика запусков SpaceX по сравнению с конкурентами» или «Экспорт нефти РФ»).

Этап 1: Сбор. Написать скрейпер (parser) на Python (библиотеки requests, BeautifulSoup) или использовать API сайта (если есть), чтобы собрать таблицу данных.

Этап 2: Очистка. Данные всегда грязные. Нужно обработать пропуски, привести валюты к единому виду, преобразовать даты.

Этап 3: Визуализация. Построить сложный график (тепловую карту корреляций, пузырьковую диаграмму или анимированную карту изменений) с помощью библиотек matplotlib, plotly или seaborn.

Результат: Не просто вывод цифр, а дашборд (панель мониторинга), позволяющий сделать экономический или исторический вывод на основе собранных данных.

Методические рекомендации:

Для таких задач учитель выступает не проверяющим, а консультантом. Оцениваться должны:

- Процесс исследования: Как ученик искал информацию об API или формате файла.
- Оптимальность кода: Использование правильных структур данных (словари для частотности работают быстрее вложенных циклов).
- Рефлексия: Небольшое эссе ученика о том, какие трудности возникли при сборе данных и почему выбранный алгоритм оказался лучше других.

3.3. Рекомендации администрации ОО, ИПК / ИРО, иным организациям, реализующим программы профессионального развития учителей по совершенствованию преподавания учебного предмета

3.3.1. 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

Основываясь на типичных затруднениях обучающихся и профессиональных предметных и методических дефицитах учителей, можно сформировать актуальную повестку для заседаний школьных методических объединений (ШМО) учителей информатики.

Ниже представлены темы, разделенные по форматам работы: от разбора конкретных дефицитов до обмена готовыми педагогическими инструментами.

Блок 1. Методический практикум: «Работа с гетерогенными группами»

Цель: Обмен опытом преодоления разрыва между сильными и слабыми учениками в одном классе.

- Тема: «От таблицы к коду: обучение слабых учеников алгоритмизации».

Обсуждение: Как уйти от страха перед переменными? Разбор приемов «Светофор», «Таблица значений» и жестких регламентов заполнения трассировочных таблиц).

Практика: Коллеги приносят свои варианты задач-»разминок» на заполнение таблиц, которые гарантированно не пугают слабый класс.

- Тема: «Дифференциация внутри одной задачи».

Обсуждение: Как одно задание может быть базовым для одних («найди сумму») и исследовательским для других («оптимизируй поиск через битовые маски»).

Обмен банком задач: Презентация тех самых межпредметных и творческих заданий (генерация ASCII, биоинформатика, шифрование), которые выдаются сильным ученикам после выполнения основной части КИМа.

Блок 2. Мастерская контента: «Инструменты визуализации и ИИ»

Цель: Использование технологий для снижения нагрузки учителя и повышения наглядности.

- Тема: «ИИ как напарник учителя: генерация дидактики».

Обсуждение: Создание банка промптов. Как за минуту получить 5 вариантов контрольной работы по теме «Системы счисления», где каждый вариант имеет свой уровень сложности, но проверяет одни и те же навыки.

Демонстрация: Быстрая генерация изображений для задач по моделированию или условий творческих квестов.

- Тема: «Динамическая геометрия и симуляции».

Обзор: Сравнение GeoGebra, Desmos и российских аналогов. Обмен готовыми интерактивными файлами: модель вращения прямоугольника вокруг оси, наглядная демонстрация того, почему кратчайшее расстояние - перпендикуляр, а сумма углов треугольника равна 180 градусам.

Блок 3. Углубленное изучение предмета: «Алгоритмы вне учебника»

Цель: Повышение предметной компетенции учителей для подготовки олимпиадников.

- Тема: «Символьные вычисления и регулярные выражения».

Разбор: Задачи типа «Морской бой: поиск координат», «Статистический анализ текста» и «Шифр Цезаря через коды символов». Учителя делятся своими решениями на Python/Паскале, обсуждая эффективность методов find(), срезов и словарей.

- Тема: «Битовая арифметика для начинающих».

Фокус: Как объяснить операции AND, OR, XOR без сложной теории? Использование аналогий с переключателями и масками прав доступа.

Задание: Решение задачи «Анализ регистров датчика» (выделение флагов из целого числа) разными способами.

Блок 4. Трансформация оценки качества

Цель: Адаптировать систему оценивания под новые реалии (включая влияние ИИ).

- Тема: «Чек-лист вместо отметки: культура самоконтроля».

Обсуждение: Внедрение рубрикаторов («Критерии пиратского чек-листа»). Оценка работ не по количеству ошибок, а по количеству выполненных пунктов ТЗ.

Опыт: Как организовать эффективную взаимопроверку кода (реверсивное ревью), чтобы ученики сами находили баги друг у друга.

- Тема: «Проектная деятельность: Информатика + ...?»

Круглый стол: Как убедить учителя литературы принять проект «Инфографика связей персонажей», а биолога — «Анализ экологических данных».

Продукт: Создание универсального шаблона метапредметного проекта, который учитель информатики может предложить любому предметнику.

Для реализации этих тем лучше использовать активные формы вместо докладов:

- «Педагогическая мастерская»: Каждый приносит один готовый раздаточный материал (карточку-памятку, шаблон Excel, кусок кода). За час создается общий банк ресурсов объединения.

- «Аквариум»: Двое коллег публично решают сложную задачу на доске/экране, проговаривая свои мысли вслух, остальные анализируют их ход мыслей и предлагают альтернативные пути.

- Вертушка «Бинго ошибок»: Учителя выписывают самые частые ошибки своих классов на стикерах, перемешивают и разбирают всей группой, придумывая методы профилактики для каждой ошибки.

3.3.2. 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

На основе анализа результатов ЕГЭ по информатике предлагаются следующие направления повышения квалификации для учителей информатики.

Блок 1. Долгосрочные модульные курсы

Целевая аудитория: Учителя со стажем до 5 лет; педагоги классов с выраженной предметной дифференциацией.

- Название курса: «Алгоритмы без абстракции: методика обучения слабых учеников программированию».

Цель: Устранение страха перед переменными, циклами и системами счисления.

Модули:

1) Трассировка как инструмент самоконтроля: жесткая работа с таблицами значений («Дневник одной ошибки», «Метод закрытой крышки»).

2) Геймификация систем счисления: визуальные модели («Морской бой» для двоичной системы, цветовые коды).

3) Паттерны IPO (Input – Process – Output): обучение структурному написанию кода в 10–15 строк.

4) Отладка без паники: разбор типичных ошибок (индексация, типы данных int/str) по логам выполнения.

- Название курса: «Символьная информатика и обработка больших данных».

Цель: Переход от численных расчетов к работе со строками, файлами и реальными данными.

Модули:

1) Методы работы со строками (find, split, срезы, замена) на примере лингвистического и криптографического анализа.

2) Основы очистки данных: чтение CSV/JSON, борьба с «мусором» в ячейках, функции приведения типов.

3) Частотный анализ текста (задачи типа «Статистический анализ») и основы регулярных выражений.

Блок 2. Краткосрочные интенсивы и практикумы

Целевая аудитория: Опытные учителя, готовящие олимпиадников или внедряющие новые технологии.

- Формат-практикум: «Битовая арифметика и магия масок».

Фокус: Решение задач на эффективность (скорость выполнения) и ограничения памяти.

Содержание: Операции &, |, ^, <<. Реализация обмена значениями без третьей переменной. Работа с флагами состояния (как в микроконтроллерах). Задача «Пароль-супергерой» (смешанный алфавит разной мощности).

- Лаборатория: «Python для обработки естественного языка».

Фокус: Создание межпредметных проектов.

Содержание: Написание генератора стихов (цепи Маркова), анализатора настроения текста или дешифратора шифра Цезаря с автоматическим подбором ключа.

- Практикум: «ИИ как напарник программиста».

Фокус: Использование нейросетей для создания дидактики.

Содержание: Генерация уникальных вариантов заданий №6, №14, №23 ЕГЭ по информатике. Создание Unit-тестов для автоматической проверки работ учеников.

Блок 3. Форматы наставничества и тьюторского сопровождения

Целевая аудитория: Молодые специалисты; школы с низкими результатами ОГЭ/ЕГЭ.

- «Предметная клиника кода»:

Разбор реальных фрагментов неработающего кода учеников. Эксперт показывает учителю, как задавать наводящие вопросы вместо прямого исправления (Socratic method).

- Серия консультаций «От идеи к проекту»:

Помощь учителям в оформлении исследовательских работ школьников. Как перевести бытовую идею («Почему мой телефон греется?») в измеримый ИТ-проект (замеры температуры CPU, построение графиков зависимости нагрузки от времени).

- Методический аудит рабочих программ:

Корректировка КТП с учетом обязательного включения тем по информационной безопасности, алгоритмизации и обработке символьной информации, которые часто выпадают из-за нехватки времени.

Блок 4. Цифровые образовательные ресурсы (ЦОР) и асинхронное обучение

Для масштабирования опыта организациям рекомендуется создать готовые продукты:

- Банк интерактивных тренажеров «Трасса-ПРО». Веб-сервис, где ученик пошагово заполняет таблицу значений переменных прямо в браузере, а система дает мгновенную обратную связь при ошибке в расчетах.
- Конструктор межпредметных кейсов. Библиотека готовых сценариев уроков: «Информатика + Биология (анализ ДНК-последовательностей)», «Информатика + Литература (анализ текстов классиков)», «Информатика + Экономика (базовая аналитика продаж в Excel/Python)».
- Библиотека эталонных решений повышенной сложности. Видеоразборы задач творческого характера (генерация ASCII-арта, фракталы) с подробным комментарием эксперта о выборе структуры данных и оптимизации алгоритма.

3.3.3. Рекомендации по другим направлениям (при наличии)

- *Администрациям образовательных организаций*

Принять на уровне образовательной организации управленческие решения, направленные на повышение качества образования, в том числе:

- провести анализ потребности педагогов в методической поддержке по вопросам дифференциации обучения информатике;
- организовать выявление лучших практик педагогов по организации дифференцированного обучения информатике;
- организовать трансляцию лучших практик через заседания методических объединений, семинары, практикумы, мастер-классы;
- организовать участие педагогов в методических мероприятиях ГОУДПО «Коми республиканский институт развития образования», заседаниях республиканских методических объединений учителей информатики;
- организовать разработку индивидуальных образовательных маршрутов для педагогов с привлечением регионального методического актива и тьюторов Центра непрерывного повышения профессионального мастерства педагогических работников ГОУДПО «Коми республиканский институт развития образования»;
- подключить обучающихся и педагогов к проекту ГОУДПО «Коми республиканский институт развития образования» «Личный кабинет обучающегося по подготовке к ГИА» и контролировать их активность.

СОСТАВИТЕЛИ ОТЧЕТА по учебному предмету: **ИНФОРМАТИКА**

Специалисты, привлекаемые к анализу результатов ЕГЭ по учебному предмету

Фамилия, имя, отчество	*Место работы, должность, ученая степень, ученое звание, принадлежность специалиста (к региональным организациям развития образования, к региональным организациям повышения квалификации работников образования, к региональной ПК по учебному предмету, пр.)*
Дурягин Александр Максимович	ГОУ РК «Физико-математический лицей-интернат», учитель информатики, председатель республиканской предметной комиссии по проверке экзаменационных работ при проведении государственной итоговой аттестации по образовательным программам среднего общего образования и единого государственного экзамена по информатике
Габова Марина Анатольевна	ГОУДПО «КРИРО», проректор по научно-методической работе, к.п.н, доцент, региональная организация развития образования

Специалисты, привлекаемые к проведению методического анализа результатов ЕГЭ по учебному предмету и рекомендаций для региональной системы образования

Фамилия, имя, отчество	*Место работы, должность, ученая степень, ученое звание, принадлежность специалиста (к региональным организациям развития образования, к региональным организациям повышения квалификации работников образования, к региональной ПК по учебному предмету, пр.)*
…	
Габова Марина Анатольевна	ГОУДПО «КРИРО», проректор по научно-методической работе, к.п.н, доцент, региональная организация развития образования

Ответственный специалист в Республике Коми по вопросам организации проведения анализа результатов ЕГЭ по учебным предметам

Фамилия, имя, отчество	*Место работы, должность, ученая степень, ученое звание*
Афанасьева Светлана Александровна	ГАУ РК «РИЦОКО», заместитель директора по оценке качества образования