ГЛАВА 2.
Методический анализ результатов ОГЭ
по ____ИНФОРМАТИКЕ____

РАЗДЕЛ 1. ХАРАКТЕРИСТИКА УЧАСТНИКОВ ОГЭ ПО УЧЕБНОМУ ПРЕДМЕТУ

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

Таблица 2-1

Экзамен	2024 г.		2025 г.		2026 г.	
	чел.	% от общего числа участников	чел.	% от общего числа участников	чел.	% от общего числа участников
ОГЭ	4772	49,31	4748	48,19	4889	51,17
ГВЭ-9	1	0,25	2	0,41	0	0,00

1.2.Процентное соотношение юношей и девушек, участвующих в ОГЭ (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	1878	39,35	1792	37,74	1898	38,82
Мужской	2894	60,65	2956	62,26	2991	61,18

1.3. Количество участников ОГЭ по учебному предмету по категориям

Таблица 2-3

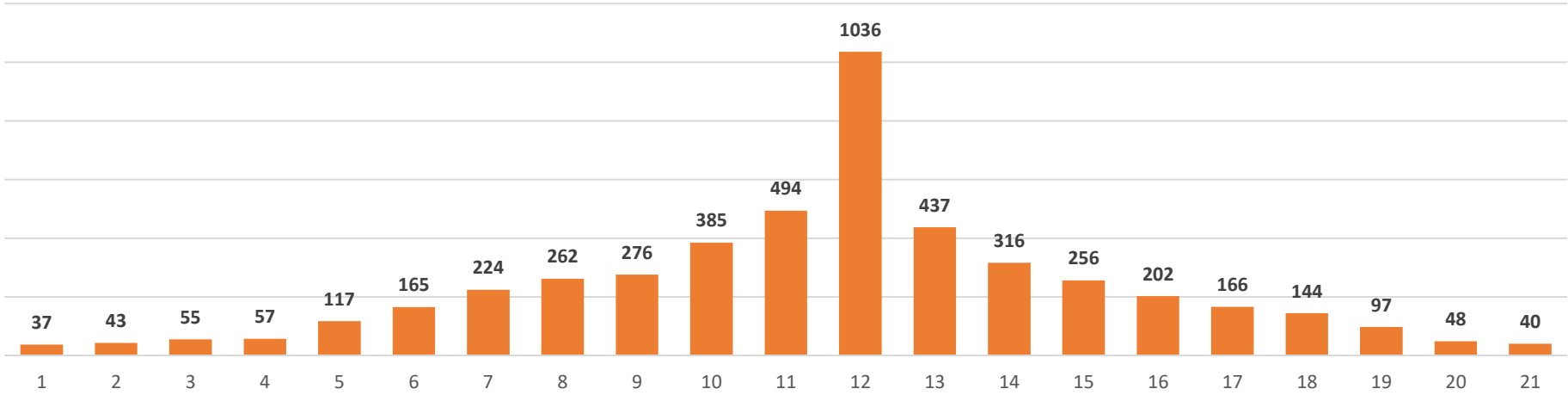
№ п/п	Участники ОГЭ	2024 г.		2025 г.		2026 г.	
		чел.	%	чел.	%	чел.	%
1.	Обучающиеся ООШ	177	3,71	188	3,96	152	3,11
2.	Обучающиеся СОШ	3072	64,38	3059	64,43	3184	65,13
3.	Обучающиеся СОШ с УИОП	785	16,45	809	17,04	850	17,39
4.	Обучающиеся гимназий	363	7,61	299	6,30	344	7,04
5.	Обучающиеся лицеев	375	7,86	393	8,28	359	7,34

Наблюдается положительная динамика количества участников ОГЭ по информатике: с 4772 чел. в 2024 г. до 4889 чел. в 2026 г. (прирост 2,5%). Основную долю участников составляют обучающиеся СОШ (65,13% в 2026 г.), при этом доля обучающихся гимназий и лицеев остаётся стабильной (14,38% в сумме). Доля участников ГВЭ-9 сократилась до 0% в 2026 г.

РАЗДЕЛ 2. ОСНОВНЫЕ РЕЗУЛЬТАТЫ ОГЭ ПО ПРЕДМЕТУ

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

(количество участников, получивших тот или иной тестовый балл)



2.2. Динамика результатов ОГЭ по предмету

Таблица 2-4

Получили отметку	2024 г.		2025 г.		2026 г.	
	чел.	%	чел.	%	чел.	%
«2»	491	10,29	415	8,74	224	4,58
«3»	2161	45,28	1857	39,11	1429	29,23
«4»	1566	32,82	1934	40,73	2741	56,06
«5»	554	11,61	542	11,42	495	10,12

2.3.Результаты ОГЭ по АТЕ региона

Таблица 2-5

№ п/п	АТЕ	Всего участников	«2»		«3»		«4»		«5»	
			чел.	%	чел.	%	чел.	%	чел.	%
1.	МО ГО «Сыктывкар»	1810	93	5,14	535	29,56	985	54,42	197	10,88
2.	МО «Воркута» РК	267	0	0,00	64	23,97	168	62,92	35	13,11
3.	МО «Вуктыл»	70	1	1,43	10	14,29	51	72,86	8	11,43
4.	МО «Инта»	166	7	4,22	35	21,08	108	65,06	16	9,64
5.	МР «Печора»	266	1	0,38	78	29,32	172	64,66	15	5,64
6.	МР «Сосногорск»	230	0	0,00	64	27,83	152	66,09	14	6,09
7.	МО «Усинск» РК	313	6	1,92	72	23,00	204	65,18	31	9,90
8.	МО «Ухта»	709	23	3,24	186	26,23	408	57,55	92	12,98
9.	МР «Ижемский»	97	4	4,12	38	39,18	44	45,36	11	11,34
10.	МР «Княжпогостский»	83	6	7,23	32	38,55	44	53,01	1	1,20
11.	МР «Койгородский»	40	4	10,00	16	40,00	20	50,00	0	0,00
12.	МО МР «Корткеросский»	105	17	16,19	50	47,62	37	35,24	1	0,95
13.	МР «Прилузский»	112	13	11,61	49	43,75	49	43,75	1	0,89
14.	МР «Сыктывдинский»	124	5	4,03	43	34,68	67	54,03	9	7,26
15.	МР «Сысольский»	62	4	6,45	25	40,32	31	50,00	2	3,23
16.	МР «Троицко-Печорский»	44	6	13,64	20	45,45	17	38,64	1	2,27
17.	МР «Удорский»	51	10	19,61	25	49,02	16	31,37	0	0,00
18.	МР «Усть-Вымский»	80	14	17,50	24	30,00	39	48,75	3	3,75
19.	МР «Усть-Куломский»	96	5	5,21	34	35,42	47	48,96	10	10,42
20.	МР «Усть-Цилемский»	83	4	4,82	21	25,30	51	61,45	7	8,43
21.	ГПОУ «Гимназия искусств при Главе Республики Коми»	18	0	0,00	1	5,56	14	77,78	3	16,67
22.	ГОУ «КРЛ при СГУ» г. Сыктывкар	14	0	0,00	0	0,00	3	21,43	11	78,57
23.	ГОУ РК «ФМЛИ»	31	0	0,00	0	0,00	9	29,03	22	70,97

№ п/п	АТЕ	Всего участников	«2»		«3»		«4»		«5»	
			чел.	%	чел.	%	чел.	%	чел.	%
	г. Сыктывкар									
24.	ГОУ РК «ШИ № 1» г. Воркута	9	1	11,11	7	77,78	1	11,11	0	0,00
25.	ГОУ «Лицей для одаренных детей»	9	0	0,00	0	0,00	4	44,44	5	55,56

2.4. Результаты по группам участников экзамена с различным уровнем подготовки с учетом типа ОО

Таблица 2-6

№ п/п	Участники ОГЭ	Доля участников, получивших отметку					
		«2»	«3»	«4»	«5»	«4» и «5» (качество обучения)	«3», «4» и «5» (уровень обученности)
1.	ООШ	7,89	39,47	48,03	4,61	52,63	92,11
2.	СОШ	5,31	31,97	56,31	6,41	62,72	94,69
3.	СОШ с УИОП	4,35	30,59	58,47	6,59	65,06	95,65
4.	Гимназии	1,16	16,86	66,86	15,12	81,98	98,84
5.	Лицеи	0,56	9,19	41,23	49,03	90,25	99,44

2.5. Выделение перечня ОО, продемонстрировавших наиболее высокие результаты ОГЭ по предмету

Таблица 2-7

№ п/п	Название ОО	Доля участников, получивших отметку «2»	Доля участников, получивших отметки «4» и «5» (качество обучения)	Доля участников, получивших отметки «3», «4» и «5» (уровень обученности)
1.	МАОУ «Технологический лицей» г. Сыктывкар	0,00	97	100,00
2.	МАОУ «ЛНД» г. Сыктывкар	0,00	100	100,00
3.	МАОУ «Лицей № 1»	0,00	96	100,00

№ п/п	Название ОО	Доля участников, получивших отметку «2»	Доля участников, получивших отметки «4» и «5» (качество обучения)	Доля участников, получивших отметки «3», «4» и «5» (уровень обученности)
	г. Сыктывкара			
4.	МАОУ «Гимназия при Главе МР «Сосногорск»	0,00	100	100,00
5.	МОУ «Лицей №1» г. Ухта	0,00	100	100,00
6.	МАОУ «УТЛ» г. Ухта	0,00	100	100,00
7.	ГОУ «КРЛ при СГУ» г. Сыктывкар	0,00	100	100,00
8.	ГПОУ «Гимназия искусств при Главе Республики Коми»	0,00	94	100,00
9.	ГОУ РК «ФМЛИ» г. Сыктывкар	0,00	100	100,00

2.6. Выделение перечня ОО, продемонстрировавших самые низкие результаты ОГЭ по предмету

Таблица 2-8

№ п/п	Название ОО	Доля участников, получивших отметку «2»	Доля участников, получивших отметки «4» и «5» (качество обучения)	Доля участников, получивших отметки «3», «4» и «5» (уровень обученности)
1.	МОУ «Усогорская «СОШ с УИОП» (Удорский р-н)	25,93	18,52	74,07
2.	МБОУ «СОШ № 2» г. Микунь (Усть-Вымский р-н)	27,27	36,36	72,73
3.	МБОУ «СОШ № 3» пгт. Жешарт (Усть-Вымский р-н)	38,46	38,46	61,54

2.7.ВЫВОДЫ о характере результатов ОГЭ по предмету в 2026 году и в динамике

В 2026 году наблюдается значительное улучшение результатов ОГЭ по информатике: доля неудовлетворительных отметок снизилась с 10,29% (2024) до 4,58% (2026), доля отметок «4» и «5» выросла с 44,43% до 66,18%. Качество обучения составило 66,18%, уровень обученности – 95,42%. Наиболее высокие результаты показали гимназии (качество 81,98%) и лицеи (90,25%). Среди АТЕ лидируют Воркута (13,11% отметок «5»), Ухта (12,98%), Вуктыл (11,43%) и специализированные ОО. Наибольшие затруднения в Удорском (19,61%), Корткеросском (16,19%) и Усть-Вымском (17,50%) районах.

Раздел 3. МЕТОДИЧЕСКИЙ АНАЛИЗ РЕЗУЛЬТАТОВ ВЫПОЛНЕНИЯ ЗАДАНИЙ КИМ 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. 2-9. 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. 2-10.

Таблица 2-9

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Средний процент выполнения[1]	Процент выполнения задания в Республике Коми в группах участников экзамена, получивших отметку			
				«2»	«3»	«4»	«5»
1.	Оценивать объём памяти, необходимый для хранения текстовых данных	Б	90,63	25,00	83,55	98,10	99,39
2.	Уметь декодировать кодовую последовательность	Б	90,04	51,34	84,46	95,04	95,96
3.	Определять истинность составного высказывания	Б	82,76	13,39	68,02	93,43	97,58
4.	Анализировать простейшие модели объектов	Б	85,48	21,43	73,48	94,67	98,18
5.	Анализировать простые алгоритмы для конкретного исполнителя с фиксированным набором команд	Б	87,95	21,43	79,50	95,88	98,59

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Средний процент выполнения[1]	Процент выполнения задания в Республике Коми в группах участников экзамена, получивших отметку			
				«2»	«3»	«4»	«5»
6.	Формально исполнять алгоритмы, записанные на языке программирования	Б	71,67	6,25	46,47	86,28	93,13
7.	Знать принципы адресации в сети Интернет	Б	85,72	24,55	72,15	95,77	96,97
8.	Понимать принципы поиска информации в Интернете	П	69,36	8,04	41,64	84,64	92,53
9.	Умение анализировать информацию, представленную в виде схем	П	81,80	12,50	61,79	95,04	97,58
10.	Записывать числа в различных системах счисления	Б	76,29	7,59	49,97	91,57	98,79
11.	Поиск информации в файлах и каталогах компьютера	Б	79,85	15,18	53,11	95,70	98,59
12.	Определение количества и информационного объёма файлов, отобранных по некоторому условию	Б	76,56	13,84	48,57	92,56	97,17
13.	Создавать презентации (вариант задания 13.1) или создавать текстовый документ (вариант задания 13.2)	П	27,45	3,79	11,86	28,62	76,67
14.	Умение проводить обработку большого массива данных с использованием средств электронной таблицы	В	17,72	0,45	3,17	14,74	84,04
15.	Создавать и выполнять программы для заданного исполнителя	В	27,39	0,45	6,89	28,31	93,64
16.	Создавать и выполнять программы на универсальном языке программирования	В	4,17	0,00	0,07	0,86	36,26
Всего заданий – **16**; из них по типу заданий: с кратким ответом – **12**, с развёрнутым ответом – **4**.							

Номер задания в КИМ	Проверяемые элементы содержания / умения	Уровень сложности задания	Средний процент выполнения[1]	Процент выполнения задания в Республике Коми в группах участников экзамена, получивших отметку			
				«2»	«3»	«4»	«5»
по уровню сложности: Б – **10**; П – **3**; В – **3**. Максимальный первичный балл за работу – **21**. Общее время выполнения работы – **2 часа 30 минут (150 мин.)**.							

Таблица 2-10

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 экзамен, получивших отметку			
		«2»	«3»	«4»	«5»
1-1	0	75	16,45	1,9	0,61
1-1	1	25	83,55	98,1	99,39
1-2	0	48,66	15,54	4,96	4,04
1-2	1	51,34	84,46	95,04	95,96
1-3	0	86,61	31,98	6,57	2,42
1-3	1	13,39	68,02	93,43	97,58
1-4	0	78,57	26,52	5,33	1,82
1-4	1	21,43	73,48	94,67	98,18
1-5	0	78,57	20,5	4,12	1,41
1-5	1	21,43	79,5	95,88	98,59
1-6	0	93,75	53,53	13,72	6,87
1-6	1	6,25	46,47	86,28	93,13
1-7	0	75,45	27,85	4,23	3,03
1-7	1	24,55	72,15	95,77	96,97
1-8	0	91,96	58,36	15,36	7,47
1-8	1	8,04	41,64	84,64	92,53
1-9	0	87,5	38,21	4,96	2,42
1-9	1	12,5	61,79	95,04	97,58
1-10	0	92,41	50,03	8,43	1,21
1-10	1	7,59	49,97	91,57	98,79
1-11	0	84,82	46,89	4,3	1,41

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 экзамен, получивших отметку			
		«2»	«3»	«4»	«5»
1-11	1	15,18	53,11	95,7	98,59
1-12	0	86,16	51,43	7,44	2,83
1-12	1	13,84	48,57	92,56	97,17
2-13	0	92,41	78,03	54,51	2,22
2-13	1	7,59	20,22	33,75	42,22
2-13	2	0	1,75	11,75	55,56
2-14	0	99,11	93,21	73,51	0,4
2-14	1	0,45	4,41	12,62	6,87
2-14	2	0,45	2,03	10	32,93
2-14	3	0	0,35	3,87	59,8
2-15	0	99,55	91,39	68,81	4,04
2-15	1	0	3,43	5,76	4,65
2-15	2	0,45	5,18	25,43	91,31
2-16	0	100	99,93	98,91	60,81
2-16	1	0	0	0,47	5,86
2-16	2	0	0,07	0,62	33,33

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

Таблица 2-3

	в группе получивших отметку «2»	в группе получивших отметку «3»	в группе получивших отметку «4»	в группе получивших отметку «5»
1	2	3	4	5
Наиболее успешно выполненные задания **базового** уровня сложности	Задание 2 (51,34%), задание 1 (25,00%), задание 7 (24,55%)	Задание 2 (84,46%), задание 1 (83,55%), задание 5 (79,50%)		

Наименее успешно выполненные **задания** базового уровня сложности	Задание 6 (6,25%), задание 10 (7,59%), задание 12 (13,84%)	Задание 6 (46,47%), задание 10 (49,97%), задание 12 (48,57%)	Задание 6 (86,28%), задание 10 (91,57%), задание 12 (92,56%)	
Наиболее успешно выполненные задания **повышенного** уровня сложности	Задание 9 (12,50%), задание 8 (8,04%)	Задание 9 (61,79%), задание 8 (41,64%)	Задание 9 (95,04%), задание 8 (84,64%)	Задание 9 (97,58%), задание 8 (92,53%)
Наименее успешно выполненные задания **повышенного** уровня сложности		Задание 8 (41,64%), задание 9 (61,79%)	Задание 8 (84,64%), задание 9 (95,04%)	Задание 8 (92,53%), задание 9 (97,58%)
Наиболее успешно выполненные задания **высокого** уровня сложности		Задание 13 (11,86%), задание 15 (6,89%)	Задание 13 (28,62%), задание 15 (28,31%)	Задание 15 (93,64%), задание 14 (84,04%)
Наименее успешно выполненные задания **высокого** уровня сложности			Задание 16 (0,86%), задание 14 (14,74%)	Задание 16 (36,26%), задание 13 (76,67%)

3.1.2. Прочие результаты статистического анализа (при наличии)

Типичные ошибки, возможные причины ошибок (на основе веера ответов):

Задание 1 – на оценивание объёма памяти, необходимого для хранения текстовых данных. Выполняемость задания составила 90,63%. Задание выполнено на ожидаемом уровне. Типичная ошибка: неумение переводить из одной единицы измерения информации в другую.

Задание 2 – на умение декодировать кодовую последовательность. Выполняемость задания составила 90,04%. Задание выполнено на ожидаемом уровне. Типичная ошибка: невнимательное прочтение текста. Учащиеся выполняют задание на «автомате» и не дочитывают постановку вопроса.

Задание 3 – на определение истинности составного высказывания. Выполняемость задания составила 82,76%. Задание выполнено на ожидаемом уровне. Типичная ошибка: невнимательное прочтение текста, путаница с приоритетом выполнения логических операций.

Задание 4 – на анализ простейших моделей объектов. Выполняемость задания составила 85,48%. Задание выполнено на ожидаемом уровне. Типичная ошибка: невнимательное прочтение текста.

Задание 5 – на анализ простых алгоритмов для конкретного исполнителя с фиксированным набором команд. Выполняемость задания составила 87,95%. Задание выполнено на ожидаемом уровне. Возможной ошибкой является невнимательный счет.

Задание 6 – на формальное исполнение алгоритмов, записанных на языке программирования. Выполняемость задания составила 71,67%. Типичные ошибки: незнание основных алгоритмических конструкций, незнание логических операций, невнимательное прочтение текста.

Задание 7 – на знание принципов адресации в сети Интернет. Выполняемость задания составила 85,72%. Задание выполнено на ожидаемом уровне. Возможная ошибка – невнимательное выписывание ответа. Кроме того, учащиеся часто при записи адреса файла в сети Интернет сперва указывают имя файла, а затем имя сервера.

Задание 8 – на понимание принципов поиска информации в Интернете. Выполняемость задания составила 69,36%. Типичные ошибки: вычислительные ошибки, отсутствие знаний по теории множеств, отсутствие знаний по алгебре логики.

Задание 9 – на умение анализировать информацию, представленную в виде схем. Выполняемость задания составила 81,80%. Задание выполнено на ожидаемом уровне. Типичная ошибка: невнимательный подсчет.

Задание 10 – на запись чисел в различных системах счисления. Выполняемость задания составила 76,29%. Типичные ошибки: вычислительные ошибки, ошибки в алгоритмах перевода из одной системы счисления в другую.

Задание 11 – на поиск информации в файлах и каталогах компьютера. Выполняемость задания составила 79,85%. Типичные ошибки: невнимательное прочтение текста, отсутствие навыка поиска информации в текстовом файле. Учащиеся, не выполнившие задание, чаще всего, никогда не пользовались инструментом поиска информации до экзамена.

Задание 12 – на определение количества и информационного объёма файлов, отобранных по некоторому условию. Выполняемость задания составила 76,56%. Типичные ошибки: невнимательное прочтение текста, незнание инструментов поиска файлов в операционных системах Windows и Linux.

Задание 13 – на умение создавать презентации (вариант задания 13.1) или создавать текстовый документ (вариант задания 13.2). Выполняемость задания составила 27,45%. Задание выполнено на ожидаемом уровне. Типичные ошибки: невнимательное прочтение текста, отсутствие навыка работы с редактором презентаций, не соответствие презентации макету, искажение изображений, неверный размер шрифта, отсутствие единого стиля текста, отсутствие навыка работы с текстовым процессором, отсутствие выравнивание ячеек таблицы по вертикали или по горизонтали, отсутствие выравнивания текста по ширине, отсутствие отступа красной строки, отсутствие оформления текста (в соответствие образцу). Среди новых типичных ошибок можно выявить сохранение файла в неверном формате, отличном от формате, указанном в задании.

Задание 14 – на умение проводить обработку большого массива данных с использованием средств электронной таблицы. Выполняемость задания составила 17,72%. Типичные ошибки: отсутствие навыка работы с электронными таблицами, невнимательное прочтение текста, неверное использование встроенных функций, ошибки при сокращении чисел (по правилам математического округления), отсутствие легенды в диаграмме, отсутствие подписи данных на секторах диаграммы. Необходимо обратить внимание на решение задания с использованием средств электронной таблицы, в котором расчет требует знания таких понятий, как процент, среднее арифметическое значение. Анализ работ показывает, что часто, используя функцию ЕСЛИ для каждой строки, учащиеся верно строят

дополнительный столбец, в котором, например, выносят значения оценок по математике, полученных учениками чей балл по информатике более 60, однако против строк с меньшими баллами ставят нули, что приводит к ошибке, если при вычислении среднего балла по математике используют стандартную функцию СРЗНАЧ. В этом случае нули тоже учитываются в расчете. Необходимо обучать учащихся пользоваться различными методами обработки: формулы, фильтры, сортировка.

Задание 15 – на умение создавать и выполнять программы для заданного исполнителя 27,39%. Типичные ошибки: невнимательное прочтение текста, создание программы для конкретной обстановки, программа не завершается, исполнитель разбивается, отсутствие понимание, что поле исполнителя бесконечно и не имеет внешних стен.

Задание 16 – на умение создавать и выполнять программы на универсальном языке программирования. Выполняемость задания составила 4,17%. Типичные ошибки: невнимательное прочтение текста, незнание основных алгоритмических конструкций, неверная проверка условий (кратность числу, окончание числа определенной цифрой).

Ниже приведены задания, которые вызвали наибольшие затруднения.

Задание №6 проверяет навыки и умения учеников по теме «Алгоритмизация и программирование». Здесь и далее будут приведены примеры заданий из открытого варианта №326.

6 Ниже приведена программа, записанная на пяти языках программирования.

Алгоритмический язык

```
алг
нач
цел s, t, A
ввод s
ввод t
ввод A
если s > A или t > 12
  то вывод "YES"
  иначе вывод "NO"
все
кон
```

Паскаль

```
var s, t, A: integer;
begin
 readln(s);
 readln(t);
 readln(A);
 if (s > A) or (t > 12)
  then
    writeln('YES')
  else
    writeln('NO')
end.
```

Бейсик

```
DIM s, t, A AS INTEGER
INPUT s
INPUT t
INPUT A
IF s > A OR t > 12 THEN
  PRINT "YES"
ELSE
  PRINT "NO"
ENDIF
```

Python

```
s = int(input())
t = int(input())
A = int(input())
if (s > A) or (t > 12):
    print("YES")
else:
    print("NO")
```

C++

```
#include <iostream>
using namespace std;

int main() {
   int s, t, A;
   cin >> s;
   cin >> t;
   cin >> A;
   if (s > A || t > 12)
     cout << "YES" << endl;
   else
     cout << "NO" << endl;
   return 0;
 }
```

Было проведено 9 запусков программы, при которых в качестве значений переменных *s* и *t* вводились следующие пары чисел:
(13, 2); (11, 12); (–12, 12); (2, –2); (–10, –10); (6, –5); (2, 8); (9, 10); (1, 13).
Укажите наибольшее целое значение параметра *A*, при котором для указанных входных данных программа напечатает «NO» пять раз.

Ответ: _____.

К типичной ошибке можно отнести недостаточное понимание учащимися принципов работы стандартных алгоритмических конструкций. В данном задании учащиеся могут перепутать логические операции. Кроме того, учащиеся иногда считают количество не тех пар (вместо ответа YES, считают ответ NO).

Задание №8 проверяет принципы поиска информации в Интернете. Данное задание связано с темами «Алгебра логики» и «Множества» из математики.

8 В языке запросов поискового сервера для обозначения логической операции «ИЛИ» используется символ «|», а для обозначения логической операции «И» – символ «&».
В таблице приведены запросы и количество найденных по ним страниц некоторого сегмента сети Интернет.

Запрос	Найдено страниц (в сотнях тысяч)
Император	68
Китай	61
Царь	33
Император | Китай | Царь	120
Император & Китай	26
Царь & Китай	0

Какое количество страниц (в сотнях тысяч) будет найдено по запросу *Царь & Император*?
Считается, что все запросы выполнялись практически одновременно, так что набор страниц, содержащих все искомые слова, не изменялся за время выполнения запросов.

Ответ: _____.

Среди типичных ошибок подобных заданий можно выделить такие как: нарушение порядка логических операций, невнимательность и ошибки в подсчетах.

Задание №10 проверяет навыки записывать числа в различных системах счисления.

10 Определите наибольшее среди чисел, записанных в двоичной, восьмеричной и шестнадцатеричной системах счисления:
100101110_2, 620_8, 181_{16}.
В ответе запишите число в десятичной системе счисления. Основание системы счисления указывать не нужно.

Ответ: _____.

В данном задании числа даны в различных системах счисления. Один из самых простых способов решения – перевод всех чисел в одну систему счисления (например, в десятичную) и проведение арифметических действий в этой системе. Типичные ошибки – незнание алгоритма перевода из произвольной системы счисления в десятичную систему счисления и невнимательное применение данного алгоритма (арифметические ошибки).

Задание 12 проверяет умение определять количество и информационного объёма файлов, отобранных по некоторому условию.

12 Сколько всего файлов с расширениями .jpeg и .txt содержится в подкаталогах **Васнецов** и **Малевич** каталога **DEMO-12/Живопись**, а также в подкаталоге **Маяковский** каталога **DEMO-12/Поэзия**? В ответе укажите только число.

Ответ: _____.

Учащиеся невнимательно читают задание, проверяют не все подкаталоги.

Задание №14 проверяет умение проводить обработку большого массива данных с использованием средств электронной таблицы. Данное задание традиционно вызывает определенные сложности среди учащихся. Помимо типичных ошибок можно отметить, что тема «электронные таблицы», скорее всего, изучается в недостаточной степени.

14 В электронную таблицу занесли данные наблюдения за погодой в течение одного года. Ниже приведены первые пять строк таблицы.

	A	B	C	D	E	F
1	Дата	Температура	Осадки	Давление	Ветер	Скорость ветра
2	1 января	0,7	15,2	748	ЮВ	4,2
3	2 января	0,4	4,6	751	В	4,7
4	3 января	–1,9	1,4	747	С	2,4
5	4 января	–7,7	0,2	752	З	4,7

В столбце A записана дата наблюдения; в столбце B – среднесуточная температура воздуха для указанной даты; в столбце C – количество выпавших осадков (в миллиметрах) для указанной даты; в столбце D – среднесуточное атмосферное давление (в миллиметрах ртутного столба). В столбце E записано направление ветра для указанной даты – одно из восьми возможных значений: «СЗ», «С», «СВ», «В», «ЮВ», «Ю», «ЮЗ», «З». В столбце F записана среднесуточная скорость ветра (в метрах в секунду).

Всего в электронную таблицу были занесены данные по всем 365 дням года в хронологическом порядке.

Откройте файл с данной электронной таблицей (расположение файла Вам сообщат организаторы экзамена). На основании данных, содержащихся в этой таблице, выполните задания.

1. Какое среднее количество осадков выпадало за сутки в осенние месяцы (сентябрь, октябрь, ноябрь)? Ответ на этот вопрос запишите в ячейку H2 таблицы.
2. Какая средняя температура была в те дни года, когда дул северный (С) ветер? Ответ на этот вопрос запишите в ячейку H3 таблицы.
3. Постройте круговую диаграмму, отображающую соотношение количества дней, когда дули ветры «В», «СВ» и «ЮВ». Левый верхний угол диаграммы разместите вблизи ячейки G6. В поле диаграммы должны присутствовать легенда (обозначение, какой сектор диаграммы соответствует каким данным) и числовые значения данных, по которым построена диаграмма.

Ответы должны быть вычислены с точностью не менее двух знаков после запятой. Полученную таблицу необходимо сохранить под именем, указанным организаторами экзамена.

В данном задании можно отметить следующие типичные ошибки: при выполнении задания 1 забываю посичтать среднее значение для конкретных месяцев. При задании 2 неверно учитывают условие «дул северный ветер» - неверно учитывают значения «СЗ» и «СВ». При выполнении задания 3 на диаграмме указывают процентные значения, а не числовые; забывают указывать подписи данных.

Задание 15 проверяет навыки создания и выполнения программы для заданного исполнителя. Типичные ошибки: невнимательное прочтение текста, создание программы для конкретной обстановки, программа не завершается, исполнитель разбивается, отсутствие понимание, что поле исполнителя бесконечно и не имеет внешних стен.

15 Исполнитель Робот умеет перемещаться по лабиринту, начерченному на плоскости, разбитой на клетки. Между соседними (по сторонам) клетками может стоять стена, через которую Робот пройти не может.
У Робота есть девять команд. Четыре команды – это команды-приказы:

вверх вниз влево вправо

При выполнении любой из этих команд Робот перемещается на одну клетку соответственно: вверх ↑, вниз ↓, влево ←, вправо →. Если Робот получит команду передвижения сквозь стену, то он разрушится.
Также у Робота есть команда **закрасить**, при которой закрашивается клетка, в которой Робот находится в настоящий момент.
Ещё четыре команды – это команды проверки условий. Эти команды проверяют, свободен ли путь для Робота в каждом из четырёх возможных направлений:

сверху свободно снизу свободно слева свободно справа свободно

Эти команды можно использовать вместе с условием «**если**», имеющим следующий вид:
если *условие* **то**
последовательность команд
все
Здесь *условие* – одна из команд проверки условия.
Последовательность команд – это одна или несколько любых команд-приказов.
Например, для передвижения на одну клетку вправо, если справа нет стенки и закрашивания клетки, можно использовать такой алгоритм:
если справа свободно то
вправо
закрасить
все
В одном условии можно использовать несколько команд проверки условий, применяя логические связки **и**, **или**, **не**, например:
если (справа свободно) и (не снизу свободно) то
вправо
все
Для повторения последовательности команд можно использовать цикл «**пока**», имеющий следующий вид:
нц пока *условие*
последовательность команд
кц
Например, для движения вправо, пока это возможно, можно использовать следующий алгоритм:

нц пока справа свободно
вправо
кц

Выполните задание

Робот находится в правой клетке узкого горизонтального коридора. Ширина коридора – одна клетка, **длина коридора может быть произвольной**. Возможный вариант начального расположения Робота приведён на рисунке (Робот обозначен буквой «Р»):



Напишите для Робота алгоритм, закрашивающий все клетки внутри коридора и возвращающий Робота в исходную позицию. Например, для приведённого выше рисунка Робот должен закрасить следующие клетки (см. рисунок):



Алгоритм должен решать задачу для произвольного конечного размера коридора. При исполнении алгоритма Робот не должен разрушиться, выполнение алгоритма должно завершиться.

Алгоритм может быть выполнен в среде формального исполнителя или записан в текстовом редакторе.

Сохраните алгоритм в формате программы Кумир или в текстовом файле. Название файла и каталог для сохранения Вам сообщат организаторы экзамена.

В данной задаче были допущены следующие ошибки: учащиеся не учитывают, что робот должен вернуться в исходную клетку; не учитывают, что длина коридора может быть произвольной (в том числе 1 клетка).

Задание 16 проверяет навыки создавать и выполнять программы на универсальном языке программирования.

16 Напишите программу подсчёта количества элементов последовательности натуральных чисел, запись которых в восьмеричной системе счисления четырёхзначна и оканчивается на цифру 7. В ответе запишите количество искомых элементов.

На вход программе сначала подаётся количество элементов последовательности N ($1 \leq N \leq 1000$), затем каждый элемент последовательности в отдельной строке.

Программа должна напечатать только одно число – количество искомых элементов последовательности.

Пример работы программы

Входные данные	Выходные данные
5 487 559 895 3031 4127	3

В данном случае к типичным ошибкам можно отнести неправильный способ проверки окончания сила на 7 именно в восьмеричной системе счисления (проверяют остаток в десятичной системе счисления); проверка четырёхзначности числа (проверяют количество разрядов в десятичной системе счисления). Некоторые ученики применяют не тот цикл (while вместо for) и алгоритм получает числа пока не будет введен 0, вместо фиксированного количества чисел.

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 (получивших отметку «2»), включая методические рекомендации для учителей

3.2.1.1. Описание предметной подготовки участников ОГЭ, получивших отметку «2», 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 (отметка «2») наиболее успешно справились с заданием 2 (декодирование кодовой последовательности) – 51,34%, что свидетельствует о частичной сформированности умения анализировать кодовые последовательности. Относительно успешно выполнены задания 1 (25,00%) и 7 (24,55%), что указывает на начальные навыки оценки объёма памяти и знаний адресации в сети Интернет.

- 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

Наименее сформированы умения: формального исполнения алгоритмов (задание 6 – 6,25%), записи чисел в системах счисления (задание 10 – 7,59%), поиска информации в Интернете (задание 8 – 8,04%). Практически полностью отсутствуют навыки работы с электронными таблицами (задание 14 – 0,45%), программирования для исполнителя (задание 15 – 0,45%) и на универсальном языке (задание 16 – 0,00%). Типичные ошибки в задании 6 связаны с неверным определением количества проходов цикла и условий ветвления; в задании 10 – с непониманием правил перевода между системами счисления; в задании 8 – с неверным применением формул включений-исключений для поисковых запросов.

Таблица 2-4

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП[2]
1.	Формальное исполнение алгоритмов	8 класс
2.	поиска информации в Интернете	7, 9 классы
3.	электронные таблицы	9 класс
4.	программирование для исполнителя	8 класс
5.	Программирование на универсальном языке	8 класс

- Реалистичные «зоны роста» в части предметной подготовки для уверенного получения отметки «3»

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

Для уверенного получения отметки «3» участникам с минимальным уровнем необходимо освоить базовые задания 1, 2, 4, 5, 7 (процент выполнения в группе «3» составляет 73–85%), а также повысить результативность по заданиям 3, 9, 11, 12 до 50–60%. Приоритетными являются: отработка навыков исполнения алгоритмов, работы с системами счисления и поиска информации в файловой системе.

3.2.1.2. Методический анализ качества освоения метапредметных результатов ФГОС участниками ОГЭ, получивших отметку «2»

- 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;

Наименее успешно выполненные обучающимися данной группы задания базового уровня сложности: задание 6 (6,25%), задание 10 (7,59%), задание 12 (13,84%).

Типичные ошибки:

- отсутствие ответа, незнание основных алгоритмических конструкций, незнание логических операций, невнимательное прочтение текста (задание 6);
- отсутствие ответа, вычислительные ошибки, ошибки в алгоритмах перевода из одной системы счисления в другую (задание 10);
- отсутствие ответа, невнимательное прочтение текста, незнание инструментов поиска файлов в операционных системах Windows и Linux (задание 12).

На успешность выполнения данных заданий повлияла несформированность познавательных универсальных действий: логических (с учетом предложенной задачи выявлять закономерности и противоречия в рассматриваемых фактах, данных и наблюдениях; выявлять причинно-следственные связи при изучении явлений и процессов; делать выводы с использованием дедуктивных и индуктивных умозаключений, умозаключений по аналогии, формулировать гипотезы о взаимосвязях), исследовательских (самостоятельно формулировать обобщения и выводы по результатам проведенного наблюдения, опыта, исследования, владеть инструментами оценки достоверности полученных выводов и обобщений), действий по работе с информацией (выбирать, анализировать, систематизировать и интерпретировать информацию различных видов и форм представления, находить сходные аргументы (подтверждающие или опровергающие одну и ту же идею, версию) в различных информационных источниках).

Обучающиеся данной группы недостаточно овладели универсальными учебными коммуникативными действиями: воспринимать и формулировать суждения, выражать себя (свою точку зрения) в письменных текстах. Об этом свидетельствуют типичные ошибки, связанные с неверным прочтением и интерпретацией текстов заданий и неверной формулировкой ответов.

Так, в задании 6 верный ответ – число 8. Среди неверных ответов были следующие: 0, 1, 10, 11, -11, 1112, 12, -1212, 13, 132, 14, 16, 2, 21111, 225, 3, 4, 5, 6, 65, 6724315, 7.

В задании 10 верный ответ – 400. Вместо этого обучающиеся записывали: в двоичной системе 1001101100, 11011001, 100, 1001010010, 110101, а также другие варианты 10153, 103, 1087, 1088, 1153, 1154, 1207, 126, 127, 1568, 16, 162, 18116, 19, 20 и др. в десятичной системе счисления.

Среди учащихся, получивших оценку «2», сравнительно неплохой результат имеют задания, которые не требуют глубоких знание предмета (знания формул, алгоритмов и т.п.), которые можно сделать перебором, повтором указанного алгоритма и т.п.

Выполнение всех заданий требует умения владеть способами самоконтроля, самомотивации и рефлексии. При недостаточной сформированности самоконтроля (как итогового, так и пошагового) и рефлексии могут возникать ошибки, описки, например, при указании имен файлов в заданиях 13,14,15.

Выполнение всех заданий требует развитого умения оценивать соответствие результата цели и условиям. Большое число ошибок возникает при недостаточной его сформированности.

Значительное количество пропусков ответов (до 4,5% в задании 6) свидетельствует о низкой самоорганизации и несформированности регулятивных УУД.

- Предположение о достигнутых и недостигнутых метапредметных результатах ФГОС

Метапредметные результаты ФГОС у участников данной группы достигнуты на минимальном уровне. Частично сформированы познавательные УУД и коммуникативные УУД (умение работать с информацией, представленной в виде кодов). Недостигнутыми остаются регулятивные УУД (целеполагание, планирование, контроль) и познавательные логические УУД (анализ, синтез, установление причинно-следственных связей), исследовательские и действия по работе с информацией.

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

Общие рекомендации:

1) усилить практическую направленность обучения – регулярно включать задания на формальное исполнение алгоритмов (трассировка) и перевод чисел между системами счисления;

2) использовать пошаговые инструкции и визуализацию при объяснении работы алгоритмов;

3) организовывать парную работу при решении задач на поиск информации;

4) проводить мини-проверки с обязательным анализом типичных ошибок;

5) уделить особое внимание формированию навыков самоконтроля – проверке полученного ответа на правдоподобность.

Приведем конкретные методические приемы и примеры заданий по проблемным аспектам.

Для учеников, испытывающих трудности с абстрактным мышлением, алгоритмы должны быть максимально «осязаемыми». Главная задача - убрать математическую терминологию на первых этапах и заменить её бытовыми действиями: рисованием стрелочек, раскрашиванием клеток или перекладыванием предметов.

Ниже приведены примеры заданий для 5–9 классов, построенные по принципу визуализации каждого шага.

Блок 1. Алгоритмы в быту (5–6 классы)

Цель: Научить понимать логику «Если..., то...» и строгое следование инструкции без пропусков шагов.

- Задание «Робот-пылесос»

Материал: Лист бумаги в клетку, где нарисована комната с разбросанными препятствиями (стулья, игрушки) и точкой финиша (розетка).

Инструкция-карточка:

Стань на старт (нарисуй человечка ручкой).

Пока перед тобой стена - поворачивай направо.

Если впереди препятствие (закрашенная клетка), поверни налево.

Сделай шаг вперед (проведи линию).

Повторяй пункты 2–4, пока не дойдешь до крестика.

Визуализация: Ученик физически водит ручкой по бумаге. Ошибка сразу видна — линия уперлась в стену.

- Задание «Магическая коробка» (Сортировка)

Материал: Набор из 10 карточек с числами от 1 до 100, две пустые коробки с надписями-условиями («Чётные» и «Нечётные»).

Алгоритм:

Возьми верхнюю карточку из стопки.

Посмотри на последнюю цифру числа.

Если цифра 0, 2, 4, 6 или 8 - положи в коробку «Чётные».

Во всех остальных случаях - в коробку «Нечётные».

Вернись к пункту 1, если карточки не закончились.

Почему это работает: Задание заменяет сложное понятие «алгоритм сортировки» простой механической манипуляцией.

Блок 2. Визуальные исполнители (6–7 классы)

Цель: Познакомить с понятием переменной и циклов через изменение состояния рисунка.

- Задание «Графический диктант для чайника» (или «Черепашка»)

Материал: Миллиметровая бумага, карандаш.

Система команд:

П - Поднять перо (двигаться, не рисуя).

О - Опустить перо (начать рисовать).

Л/П [число] - Поворот налево/направо на 90 градусов.

ВП [число] — Шаг вперед на указанное количество клеток.

Программа: О -> ВП 5 -> Л -> ВП 3 -> П -> ВП 2 -> П -> ВП 3

Результат: Буква "Ш". Для слабых учеников важно сначала дать им готовый рисунок и попросить записать алгоритм самим, а потом дать чужой алгоритм для исполнения.

- Задание «Светофор состояний» (Конечные автоматы)

Материал: Распечатанная схема кружков (красный, желтый, зеленый) со стрелочками переходов.

Визуализация: Стрелочки подписаны условиями. Например, от красного к зеленому ведет стрелка с надписью «Прошло 60 сек», от зеленого к желтому - «Прошло 40 сек».

Задача: Учитель говорит: «Прошло 10 секунд после включения красного». Ученик должен пальцем показать, на каком цвете сейчас находится светофор, двигаясь по стрелочкам за голосом учителя.

Блок 3. Игровые алгоритмы поиска (7–8 классы)

Цель: Объяснить работу поисковых алгоритмов без использования формул.

- Задание «Найди клад методом деления пополам»

Материал: Таблица чисел от 1 до 31, расположенных по порядку.

Роль ИИ: Один ученик загадывает число.

Инструкция угадывающему (строгая пошаговая):

Найди число ровно посередине таблицы (между 1 и 31). Это 16.

Спроси: «Твое число больше 16?».

Если ответ «Да», забудь про левую половину (зачеркни её карандашом). Твой новый список теперь от 17 до 31.

Снова найди середину нового списка.

Повторяй, пока не останется одно число.

Визуализация: Постепенное закрашивание ненужных частей таблицы. Ученик видит, как объем работы уменьшается вдвое на каждом шаге.

- Задание «Маршрут курьера» (Жадный алгоритм)

Материал: Карта района (можно упрощенная, нарисованная учителем) с точками магазинов А, Б, В, Г. Расстояния между ними подписаны прямо на линиях дорог.

Условие: Курьер стоит в точке А. Ему нужно забрать заказы во всех точках, потратив минимум времени.

Простой алгоритм для ученика:

Где ты стоишь? (А).

Из оставшихся точек (Б, В, Г) какая ближе всего к тебе? (Допустим, В).

Сотри точку В или зачеркни её. Теперь иди туда.

Повтори: из точки В куда ближе — в Б или Г?

Почему это работает: Это интуитивно понятная стратегия «не уходи далеко, если есть дело рядом», которая является базой многих компьютерных алгоритмов.

Блок 4. Разминка «Исправь ошибку робота» (5–9 классы)

Цель: Развитие навыка тестирования и отладки (дебаггинга). Ученикам дается заведомо неправильный алгоритм (например, рецепт бутерброда: «1. Намазать хлеб маслом. 2. Положить сыр. 3. Взять нож»). Задача - найти пропущенный шаг («Отрезать кусок сыра») или шаги, стоящие в неправильном порядке («Нельзя мазать масло на пустую доску»). Переупорядочивание карточек с шагами алгоритма отлично развивает логику без единого вычисления.

Для организации парной работы со слабыми учениками важно использовать принцип взаимодополняемости: один ученик выполняет техническую часть (печатает, кликает), а второй - аналитическую (читает вслух, ищет ключевые слова в тексте, проверяет результат). Это снимает страх перед клавиатурой и распределяет ответственность.

Ниже представлены три задания разного уровня сложности для 5–9 классов, направленные на развитие навыков фактчекинга и поиска информации.

Задание 1. «Цифровой детектив: Найди подделку» (5–6 классы)

Цель: Научить отличать факты от мнений и проверять информацию через разные источники.

Роли в паре:

«Детектив» (обычно более внимательный ученик): читает текст, подчеркивает подозрительные места.

«Оперативник» (уверенно владеющий мышкой/клавиатурой): вводит запросы в поисковик.

Сценарий: Пара получает распечатанный короткую заметку о России или родном крае, содержащую 2–3 грубые ошибки.

Например: «Самое глубокое озеро в мире - Байкал. Его глубина достигает 5 километров, а вода в нем настолько соленая, что там можно лежать как на диване. В озере живут белые акулы».

Алгоритм действий:

Прочитайте текст вдвоем.

«Детектив» выдвигает гипотезу: «Мне кажется, 5 километров - это слишком много».

«Оперативник» заходит в браузер и ищет запрос: «максимальная глубина озера Байкал в метрах».

Вместе находите правильный ответ (1642 метра) и исправляете ошибку ручкой прямо на листе.

Повторяете для остальных фактов («соленость воды», «обитатели»).

Результат: Исправленный текст с указанием ссылок, откуда взята правильная информация.

Задание 2. «Битва титанов: Сравнительная таблица» (7–8 классы)

Цель: Отработка навыка использования поисковых фильтров и структурирования данных.

Роли в паре:

«Штурман»: формулирует точные поисковые запросы (например, использует кавычки для точного совпадения).

«Архивариус»: заполняет таблицу, следит за единообразием данных (все размеры - только в мм или только в см).

Сценарий: Ученикам нужно подготовить справку для школьного музея по одной из тем. Тема выбирается исходя из интересов (игры, техника, история). Примеры тем: «Характеристики игровых приставок PS5 vs Xbox Series X» или «Технические данные танков Т-34 и "Тигр"».

Алгоритм действий:

Разделите зоны ответственности. Например, первый ученик ищет всё, что касается «скорости/мощности», второй - «цены/даты выхода».

Используйте строгий шаблон таблицы (названия колонок уже даны учителем):

Характеристика Объект А Объект Б

Год создания ... ...

Вес ... ...

Главное преимущество ... ...

Поиск должен вестись не на форумах, а в энциклопедиях или на официальных сайтах производителей. Пара должна договориться: «Мы берем цифры только из строки "Спецификации"». Результат: Заполненная сравнительная таблица без субъективных оценок (слов типа «лучше», «красивее»), готовая база данных.

Задание 3. «Маршрут выходного дня: Бюджетный планировщик» (8–9 классы)

Цель: Практическое применение сложного поиска (поиск по датам, ценам, расписанию) и критическая оценка источников.

Роли в паре:

«Логист»: отвечает за время работы музеев и расписание транспорта.

«Экономист»: считает общую стоимость поездки, ищет скидки.

Сценарий: Задача - составить план похода в музей или поездку в соседний город на субботу для двоих одноклассников. Есть жесткое ограничение: бюджет 2000 рублей и всего 4 часа свободного времени.

Алгоритм действий:

Определите точку старта (свой адрес школы или дома).

«Логист» ищет сайт конкретного местного музея. Ему нужно найти страницу «Билеты» и «График работы». Важно проверить, нет ли санитарного дня.

«Экономист» ищет способы добраться туда (стоимость автобуса/такси туда-обратно).

Этап проверки: Ученики меняются ролями. «Экономист» проверяет, правильно ли «Логист» сложил время в пути и время экскурсии (успеваете ли вы вернуться до конца 4 часов?). «Логист» проверяет расчеты денег.

Если бюджет превышен, пара должна совместно решить, чем пожертвовать (например, пойти пешком вместо такси или отказаться от аудиогида).

Результат: Пошаговый маршрут-карточка:

10:00 - Выход из дома.

10:30 - Покупка билетов (цена, ссылка на сайт).

11:00 – 12:30 - Экскурсия.

Итого потрачено: ___ руб.

Методический совет для учителя: Для слабых учеников обязательно введите правило «Один экран - два мнения». Оба ученика должны видеть монитор одновременно. Прежде чем кликнуть по ссылке, «Оперативник» обязан спросить: «Я открываю этот сайт?», а «Детектив» должен подтвердить: «Да, он выглядит надежно, тут есть логотип и контакты». Это приучает их к коллективному принятию решений и базовым правилам цифровой гигиены.

Слабые ученики часто не замечают собственных ошибок, потому что их внимание полностью поглощено процессом выполнения действия (найти нужную клавишу, вспомнить пароль, открыть файл). У них отсутствует «взгляд со стороны».

Чтобы сформировать самоконтроль, нужно перевести проверку из финала работы («сдал тетрадь - получил двойку») в постоянный процесс. Вот 5 методических приемов для уроков информатики.

1. Прием «Цветовые метки» (Визуализация статуса)

Слабым ученикам трудно словами описать проблему («у меня ничего не работает»). Цветовой код делает ошибку видимой и безопасной.

Как применять: При работе с текстами или алгоритмами на бумаге/в редакторе предложите использовать три маркера или инструмента рецензирования:

Зеленый: «Я уверен в этом шаге / здесь всё верно».

Желтый: «Я сомневаюсь / сделал по памяти, надо проверить».

Красный: «Здесь ошибка / я не знаю, как делать».

Почему это работает: Ученик вынужден остановиться и оценить свою уверенность перед тем, как двигаться дальше. Желтая пометка сигнализирует учителю: «Помоги мне именно тут», а красная позволяет признать ошибку без страха наказания. В средах программирования аналогом служат точки останова, которые подсвечивают строку до запуска кода.

2. Прием «Проговаривание соседу» (Перекрестный контроль)

Ученик физически не может прочитать вслух то, что написано у него с опечаткой, не заметив этого. Речь замедляет мышление и заставляет мозг обрабатывать информацию заново.

Как применять: Работа в парах при наборе текста или составлении блок-схем. Правило: прежде чем нажать кнопку «Выполнить» или «Сохранить», ученик должен повернуться к партнеру и медленно зачитать свой алгоритм или формулу. Партнер слушает с закрытым экраном или листом бумаги.

Пример: При записи пути к файлу C:\Мои документы\Проект слабый ученик часто пропускает двоеточие или обратный слэш. Читая это вслух, он спотыкается на месте пропуска, так как речевой аппарат требует паузы там, где нет знака препинания.

3. Прием «Чек-лист критика» (Алгоритмизация проверки)

Слабый ученик проверяет работу хаотично: если увидел одну ошибку - успокоился, остальное пропустил. Чек-лист дает жесткий маршрут обхода.

Как применять: Перед началом практической работы выдайте карточку с пунктами. Ученик ставит галочку только после визуальной сверки.

Для набора текста:

[ ] Все имена собственные начинаются с большой буквы?

[ ] Нет ли двойных пробелов?

[ ] Заканчивается ли предложение точкой?

Для Excel/Таблиц:

[ ] Начинаются ли формулы со знака =?

[ ] Не перепутаны ли русские "о" и английские "c/o"?

[ ] Выделена ли целая таблица перед созданием диаграммы?

Почему это работает: Это снимает когнитивную нагрузку. Ученику не нужно думать «что бы мне проверить», ему нужно просто следовать инструкции.

4. Прием «Метод намеренной ошибки учителя» (Рефлексивный анализ)

Ученики лучше всего находят чужие ошибки, особенно если они сделаны авторитетным взрослым.

Как применять: На этапе объяснения нового материала учитель сознательно допускает типичную ошибку слабого класса (например, забывает выделить диапазон ячеек перед копированием формулы, или путает расширение .txt и .docx).

Действие: Учитель останавливается и спрашивает: «Коллеги, мой компьютер выдает ошибку. Посмотрите на мои действия, где я нарушил логику алгоритма?». Класс ищет сбой.

Переход к самоконтролю: После разбора учитель говорит: «Вы сейчас нашли ошибку у меня. Теперь посмотрите на свои экраны. Сделайте ровно то же самое: проверьте, не повторили ли вы мою ошибку в своей работе прямо сейчас». Это создает ситуацию успеха - ученик находит баг, потому что только что научился его распознавать.

5. Прием «Стоп-кадр и прогноз» (Остановка перед действием)

Самая частая ошибка слабых учеников - импульсивное нажатие кнопок («э-э-э, нажму Enter, вдруг сработает»).

Как применять: Вводится жесткое правило «Трех секунд». Прежде чем выполнить команду (удалить папку, запустить макрос, отправить сообщение), рука должна замереть над мышкой/клавиатурой.

Вопрос-навигатор: Учитель ходит между партами и, видя занесенную руку, спрашивает: «Что произойдет в следующую секунду, если ты нажмешь эту кнопку? Опиши результат». Если ученик отвечает неуверенно («Не знаю... вроде сохранит»), команда отменяется до выяснения обстоятельств.

Почему это работает: Формируется связь «действие -> последствие». Ученик начинает предвидеть результат своего клика, что и является высшей формой технического самоконтроля.

Важное условие: При использовании этих приемов необходимо хвалить ученика не за правильный ответ, а за сам факт использования приема («Молодец, ты вовремя поставил красную метку!», «Отлично, ты заметил ошибку в моем примере быстрее всех!»). Это снижает тревожность и закрепляет привычку проверять себя.

3.2.2. Методический анализ выполнения заданий обучающимися с низким уровнем подготовки (в группе получивших отметку «3»), включая методические рекомендации для учителей

3.2.2.1. Описание предметной подготовки участников ОГЭ 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 группы «3» достаточно уверенно выполняют базовые задания: задание 2 (84,46%), задание 1 (83,55%), задание 5 (79,50%). Сформированы умения оценки объёма памяти, декодирования последовательностей, анализа простых алгоритмов для исполнителя. Уровень выполнения заданий повышенного уровня сложности составляет 41–62%, что свидетельствует о частичном освоении соответствующих тем.

- 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

В группе «3» наименьшую успешность показали задания 6 (46,47% – исполнение алгоритмов на языке программирования), 10 (49,97% – системы счисления) и 12 (48,57% – определение количества файлов по условию).

Типичные ошибки в задании 6 связаны с непониманием вложенных условий и логических операций (or, and); в задании 10 – с ошибками при переводе чисел из восьмеричной и шестнадцатеричной систем; в задании 12 – с неверным подсчётом количества файлов при сложных условиях отбора.

С заданиями высокого уровня сложности справляются единицы: задание 14 – 3,17%, задание 15 – 6,89%, задание 16 – 0,07%.

Таблица 2-13

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП
1.	Формальное исполнение алгоритмов	8 класс
2.	Системы счисления	8 класс
3.	Определение количества файлов по условию	7 класс
4.	Электронные таблицы	9 класс
5.	Программирование для исполнителя	8 класс
6.	Программирование на универсальном языке	8 класс

- Реалистичные «зоны роста» в части предметной подготовки для уверенного получения отметки «4»

Для уверенного перехода к отметке «4» необходимо: довести выполнение базовых заданий 6, 10, 12 до уровня 85–90%, освоить задание 8 (поиск в Интернете) и задание 9 (анализ схем) на уровне 80–85%, а также начать осваивать практические задания 13 (создание документов/презентаций) и 15 (программирование для исполнителя Робот).

3.2.2.2. Методический анализ качества освоения метапредметных результатов ФГОС участниками ОГЭ 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;

У участников группы «3» частично сформированы познавательные УУД, однако недостаточное развитие логических операций (анализ, синтез, сравнение) проявляется в ошибках при выполнении задания 6 (трассировка алгоритмов с условиями) и задания 10 (перевод между системами счисления). Слабая сформированность знаково-символических действий затрудняет выполнение заданий 2 и 10. Пробелы в смысловом чтении сказываются на заданиях 11 и 12, где требуется точное понимание условия и подсчёт по нескольким критериям.

На успешность выполнения заданий 6, 10, 12 повлияла несформированность познавательных универсальных действий: логических (с учетом предложенной задачи выявлять закономерности и противоречия в рассматриваемых фактах, данных и наблюдениях; выявлять причинно-следственные связи при изучении явлений и процессов; делать выводы с использованием дедуктивных и индуктивных умозаключений, умозаключений по аналогии, формулировать гипотезы о взаимосвязях), исследовательских (самостоятельно формулировать обобщения и выводы по результатам проведенного наблюдения, опыта, исследования, владеть инструментами оценки достоверности полученных выводов и обобщений), действий по работе с информацией (выбирать, анализировать, систематизировать и интерпретировать информацию различных видов и форм представления, находить сходные аргументы (подтверждающие или опровергающие одну и ту же идею, версию) в различных информационных источниках).

Обучающиеся данной группы недостаточно овладели универсальными учебными коммуникативными действиями: воспринимать и формулировать суждения, выражать себя (свою точку зрения) в письменных текстах. Об этом свидетельствуют типичные ошибки, связанные с неверным прочтением и интерпретацией текстов заданий и неверной формулировкой ответов.

Выполнение всех заданий требует умения владеть способами самоконтроля, самомотивации и рефлексии. При недостаточной сформированности самоконтроля (как итогового, так и пошагового) и рефлексии могут возникать ошибки, описки.

Выполнение всех заданий требует развитого умения оценивать соответствие результата цели и условиям. Большое число ошибок возникает при недостаточной его сформированности.

- Предположение о достигнутых и недостигнутых метапредметных результатах ФГОС

У данной группы частично сформированы познавательные УУД и коммуникативные УУД. Недостигнутыми остаются регулятивные УУД (целеполагание, планирование, контроль и коррекция результатов при выполнении многошаговых алгоритмов) и познавательные логические УУД (анализ, синтез, установление причинно-следственных связей), исследовательские и действия по работе с информацией.

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

Общие рекомендации:

1) организовать систематическую работу по формированию навыков трассировки алгоритмов с использованием таблиц значений переменных;

2) включить в практику регулярные задания на перевод чисел между двоичной, восьмеричной, десятичной и шестнадцатеричной системами счисления;

3) использовать задачи на поиск информации в файловой системе с постепенным усложнением условий отбора;

4) уделить внимание разбору заданий на поисковые запросы с применением кругов Эйлера;

5) стимулировать выполнение практических заданий 13–15 на компьютере.

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

Для слабых учеников трассировка (пошаговое выполнение алгоритма) - это лучший способ понять, что такое переменная. Главная ошибка таких детей в том, что они пытаются угадать конечный результат или выполняют действия в уме, путаясь в значениях.

Таблица значений превращает абстрактную программу в заполнение анкеты: «Что было? Что сделали? Что стало?».

Ниже представлены 5 заданий с постепенным усложнением для 5–9 классов.

Задание 1. «Слепое следование» (Линейный алгоритм)

Цель: Научить не пропускать строки и понимать команду присваивания x = x + 1.

Возраст: 5–6 классы.

Алгоритм:

```
1. a := 5
2. b := 10
3. c := a + b
4. a := a + 2
5. b := c - a
6. Вывод a, b, c
```

Инструкция: Перед вами пустая таблица. Перепишите её в тетрадь. Заполняйте строчки строго по порядку номеров сверху вниз. Если в строке написано a := a + 2, найдите колонку a, посмотрите число, которое там уже стоит, прибавьте к нему 2 и запишите новый результат в ту же колонку на этой строке.

Шаг	Команда	a	b	c
1	a := 5	**5**		
2	b := 10	5	**10**	
3	c := a + b	5	10	**15**
4	a := a + 2	**7**	10	15
5	b := c - a	7	**8**	15

Результат вывода: 7, 8, 15.

Задание 2. «Робот-обменник» (Обмен значениями через третью переменную)

Цель: Показать необходимость буферной переменной (транзита).

Возраст: 6–7 классы.

Сюжет: У вас есть две коробки. В одной 3 яблока (A), в другой 7 груш (B). Нужно поменять их местами так, чтобы в первой были груши, а во второй яблоки. Но руки у нас только две - сразу всё не удержать, нужна третья коробка (Tmp).

Алгоритм:

```
1. A := 3
2. B := 7
3. Tmp := A
4. A := B
5. B := Tmp
```

Типичная ошибка: Ученик пишет в шаге 4 в ячейку A семерку, а затем в шаге 5 пытается взять значение из A (где уже семь) и записать его в B. В итоге получается 7 и 7.

Метод контроля: Слабым ученикам можно выдать вырезанные карточки с цифрами. Пусть физически перекладывают карточку «3» из ячейки A в ячейку Tmp, прежде чем писать цифры в таблицу.

Задание 3. «Умный счетчик» (Циклы FOR / ДО)

Цель: Понять разницу между телом цикла и изменением счетчика.

Возраст: 7–8 классы.
Алгоритм:

```
1. S := 0 (S — это копилка)
2. i := 1
3. Повтори 3 раза:
   3.1. S := S + i
   3.2. i := i + 1
4. Вывод S
```

Важное правило оформления: Для циклов нужно добавлять строку «Начало итерации», чтобы ученик видел, как меняются данные при каждом круге.

№ шага	Команда	i	S
1	S := 0		**0**
2	i := 1	**1**	0
3.1 (Круг 1)	S := S + i	1	**1**
3.2 (Круг 1)	i := i + 1	**2**	1
3.1 (Круг 2)	S := S + i	2	**3**
3.2 (Круг 2)	i := i + 1	**3**	3
3.1 (Круг 3)	S := S + i	3	**6**
3.2 (Круг 3)	i := i + 1	**4**	6

Результат вывода: 6.

Задание 4. «Строгий охранник» (Условный оператор IF)
Цель: Научиться выполнять команды выборочно.

Возраст: 8–9 классы.
Алгоритм:

```
1. X := 10
2. Y := 5
3. Если X > 8, то Y := Y + 10
4. Если X < 5, то Y := Y * 2
5. Вывод Y
```

Подсказка для ученика: Когда доходишь до строки «Если...», прочитай условие вслух. Если оно верное («Да»), выполняй действие и пиши новое число в таблицу. Если неверное («Нет») - просто перепиши старое число из предыдущей строки в текущую, ничего не меняя.

Строка	Проверка условия	Действие	Значение Y
1-2	–	Начальная установка	**5**
3	10 > 8 (ДА!)	Y := 5 + 10	**15**
4	10 < 5 (НЕТ)	Ничего не делаем	**15**

Результат вывода: 15.

Задание 5. «Перевертыш» (Работа со строками)
Цель: Показать, что переменные могут хранить буквы, и научить работать с длинами строк.
Возраст: 8–9 классы.
Алгоритм:

```
1. Слово := "ОК"
2. Результат := ""
3. Длина := длина(Слово)
4. Номер := 1
5. Пока Номер <= Длина:
     5.1. Буква := буква(Слово, Номер)
     5.2. Результат := Буква + Результат
     5.3. Номер := Номер + 1
6. Вывод Результат
```

(Пояснение: команда *Буква + Результат* добавляет новую букву ПЕРЕД старым текстом).

Шаг	Команда	Слово	Результат	Номер	Длина
1-4	Начало	"ОК"	""	1	2
5.1-5.3 (1)	...	"ОК"	**"О"**	2	2
5.1-5.3 (2)	...	"ОК"	**"КО"**	3	2

Результат вывода: КО.

Методические советы для учителя:

- Цветовая дифференциация: Просите учеников выделять жирным или цветом те значения в таблице, которые изменились именно на этом шаге.
- Запрет на стирание: Трассировочная таблица - это летопись. Нельзя зачеркивать ошибки, нужно провести одну черту и написать правильное число рядом. Это учит видеть историю изменения данных.
- Игровизация: Называйте переменные именами героев (Здоровье=HP, Деньги=Gold), а операцию := называйте словом «стало». Фраза «Золото стало Золото плюс 50» воспринимается детьми гораздо легче, чем формальный синтаксис языка программирования.

Для слабых учеников главная проблема в теме «Системы счисления» - это абстрактность. Они не понимают, зачем нужны буквы (A, B, C) и почему нельзя просто написать число «10».

Чтобы обучение было успешным, нужно отказаться от заучивания сложных формул перевода через степени двойки на первом этапе и использовать визуализацию, аналогии из реальной жизни и жесткие алгоритмы-шпаргалки.

1. Прием «Калькулятор как инструмент проверки, а не решения»

Проблема: Слабый ученик сразу вводит число в калькулятор Windows, получает ответ и не понимает механику процесса. Метод: Запретите переводить числа через кнопку Bin, Dec или Hex. Разрешите использовать калькулятор только в режиме «Программирование» для одной цели: проверить свой результат в самом конце.

Визуализация: Нарисуйте на доске скриншот калькулятора и красным маркером перечеркните кнопки автоматического перевода. Оставьте только рабочие инструменты: деление нацело (div) и поиск остатка (mod), если позволяет модель, или просто столбик.

Цель: Ученик должен понимать, откуда взялась цифра, прежде чем увидеть правильный ответ.

2. Прием «Бородатый программист» (Перевод из десятичной системы в любую)

Это самый надежный алгоритм («деление уголком»), но слабые ученики путаются, что делать с частным, а что с остатком.

Метод жесткого алгоритма (Инструкция по шагам):

Возьми число.

Дели его на основание системы (на 2, 8 или 16).

Запиши остаток под числом (это будет буква нашего секретного кода).

Обведи частное (то, что получилось в результате деления).

Теперь забудь про верхнее число. Работай только с обведенным.

Повторяй, пока не получишь ноль.

Прочитай остатки снизу вверх.

Визуальная подсказка: Называйте остатки «очередью сообщений», которые компьютер отправляет нам снизу вверх. Чтобы прочитать слово правильно, мы должны начать с последней строчки.

Для шестнадцатеричной системы: Сделайте ученикам красивую ламинированную карточку-памятку прямо у края стола: 10 = A, 11 = B, 12 = C, 13 = D, 14 = E, 15 = F. Правило: «Если остаток больше 9, не пиши число, посмотри на шпаргалку».

3. Прием «Весы со степенями двойки» (Перевод ИЗ двоичной В десятичную)

Формула 1 * 2^3 + 0 * 2^2... пугает показателями степени.

Метод материальных объектов: Представьте, что у нас есть набор гирь: 1 кг, 2 кг, 4 кг, 8 кг, 16 кг, 32 кг...

Напишите на доске двоичное число, например 1011.

Подпишите под каждой цифрой вес гири справа налево (1, 2, 4, 8).

Правило: «Мы кладем гирю на весы ТОЛЬКО там, где стоит единица».

Действие: Ученики физически зачеркивают нули вместе с их гирями (2 и 8 зачеркнуты). Складывают оставшиеся: 8 + 0 + 2 + 1 = 11.

Это превращает алгебру в простую арифметику первого класса.

4. Прием «Родственные связи» (Переводы 2 ↔ 8 ↔16)

Объяснять этим ученикам общую теорию триад и тетрад сложно. Нужна мнемоника.

Метод «Группировки по парам/тройкам»:

Двоичная → Восьмеричная: «Собираем тройками». Читаем справа налево, отделяем цифры по 3 штуки. Если слева не хватает цифр до тройки — дописываем нули (как пустые места).

Двоичная → Шестнадцатеричная: «Собираем четверками». Отделяем по 4 цифры.

Таблица-словарь: Не заставляйте учить соответствие чисел. Дайте таблицу, где напротив 1010 написано -> А.

Упрощение для слабых: Сначала научите их переводить из 8-й и 16-й систем в двоичную (заменой цифры на группу из 3-х или 4-х бит), а уже потом - собирать эти длинные ряды единиц и нулей обратно в десятичное число методом «весов» из пункта 3. Это гораздо проще, чем считать большие степени.

5. Прием «Цветовое кодирование разрядов»

При записи длинных двоичных чисел слабые ученики часто теряют количество знаков или сдвигают колонку.

Метод: Используйте цветные ручки или подсветку ячеек в Excel.

Каждый четвертый разряд (граница байта) выделяйте другим цветом.

Пример: 0001 | 1110 | 1010

При переводе в 16-ю систему ученик видит: первая группа (синяя) - это одна цифра, вторая (красная) - вторая цифра. Это помогает избежать ошибок при выравнивании числа (дописывании лидирующих нулей).

6. Игровой прием «Магазин битов»

Сюжет: Вы - покупатель, компьютерный магазин продает память. Но ценники написаны в двоичной системе.

Пакет весом 101 грамм стоит 10 рублей.

Задача: Переведите всё в десятичную систему, чтобы понять, сколько граммов вы купите на свои 20 обычных рублей. Это добавляет смысл вычислениям. Без контекста «зачем мне это?» слабый ученик отключается после первой же строки примера.

3.2.3. Методический анализ выполнения заданий обучающимися со средним уровнем подготовки (в группе получивших отметку «4»), включая методические рекомендации для учителей

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

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

Участники группы «4» демонстрируют высокий уровень освоения базового содержания: задания 1 (98,10%), 5 (95,88%), 7 (95,77%), 2 (95,04%).

Уверенно выполняются задания повышенного уровня: 9 (95,04%) и 8 (84,64%).

Сформированы умения анализа алгоритмов, работы с интернет-адресацией, анализа информационных моделей и схем.

- 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

В группе «4» наименее успешно выполнены задания высокого уровня сложности: 14 (14,74% – обработка данных в электронных таблицах), 15 (28,31% – программирование для исполнителя), 13 (28,62% – создание презентаций/текста).

Среди базовых заданий наименьший результат показало задание 6 (86,28% – исполнение алгоритмов), что указывает на типичные затруднения при трассировке алгоритмов с составными условиями.

В задании 10 (91,57%) ошибки связаны с переводом чисел, в задании 12 (92,56%) – с подсчётом файлов по сложному условию.

Таблица 2-5

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП
1.	Формальное исполнение алгоритмов	8 класс
2.	Системы счисления	8 класс
3.	Определение количества файлов по условию	7 класс
4.	Создание презентаций	7 класс
5.	Электронные таблицы	9 класс
6.	Программирование для исполнителя	8 класс
7.	Программирование на универсальном языке	8 класс

- Реалистичные «зоны роста» в части предметной подготовки для уверенного получения отметки «5»

Для достижения отметки «5» необходимо: повысить результативность выполнения заданий высокого уровня – задания 14 (обработка табличных данных) и 15 (программирование для исполнителя) до 60–70%, задание 13 (создание документов) до 80–85%.

Требуется целенаправленная подготовка по программированию на универсальном языке (задание 16), которое в данной группе практически не выполняется (0,86%).

3.2.3.2. Методический анализ качества освоения метапредметных результатов ФГОС участниками ОГЭ 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;

Участники группы «4» демонстрируют достаточную сформированность познавательных УУД: успешно применяют знаково-символические средства (задания 2, 10), выполняют логические операции (задания 3, 6), осуществляют смысловое чтение (задания 11, 12).

Коммуникативные УУД (воспринимать и формулировать суждения, выражать себя (свою точку зрения) в письменных текстах) достаточно сформированы.

Недостаточно развиты регулятивные УУД, связанные с планированием последовательности действий при решении комплексных задач (задания 13–16), а также навыки работы с большими массивами данных (задание 14).

- Предположение о достигнутых и недостигнутых метапредметных результатах ФГОС

Метапредметные результаты ФГОС у данной группы в целом достигнуты.

Сформированы познавательные УУД на достаточно высоком уровне, коммуникативные УУД.

Требуют дальнейшего развития регулятивные УУД (самоорганизация при выполнении практических заданий).

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

Общие рекомендации:

1) организовать практикумы по обработке больших массивов данных в электронных таблицах с использованием встроенных функций (СРЗНАЧ, СУММЕСЛИ, СЧЁТЕСЛИ);

2) регулярно предлагать задачи на программирование для исполнителя (Робот) с постепенным усложнением;

3) ввести в практику написание программ на универсальном языке программирования (Python) для решения задач на обработку числовых последовательностей;

4) развивать навыки самооценки и взаимопроверки выполненных практических работ.

Приведем конкретные методические приемы и примеры заданий.

Для учеников 5–9 классов работа с большими массивами данных должна быть привязана к понятным и интересным им контекстам (школа, хобби, игры). Главная методическая цель таких практикумов - не просто научить писать формулу в строке ввода, а развить навык декомпозиции задачи: умение перевести вопрос с обычного языка на язык функций Excel или Google Таблиц.

Ниже приведены три примера практических заданий разного уровня сложности.

Практикум 1. «Школьная столовая: Анализ продаж» (Уровень: Базовый / 5–6 классы)

Контекст: Ученики получают таблицу со списком покупок одноклассников за неделю. Это знакомая среда, где легко проверить логику вычислений.

Исходные данные (фрагмент таблицы):

A	B	C
Дата	**Товар**	**Цена (руб)**
01.09	Булочка	45
01.09	Сок	60
02.09	Булочка	45
03.09	Пирожок	55
03.09	Сок	60
...	...	...
(Всего около 50 строк)		

Задание для учеников: Администрации школы нужно составить отчет по продажам. Используя функции электронных таблиц, ответьте на вопросы:

- Сколько всего булочек было куплено за всю неделю? (Функция СЧЁТЕСЛИ)
- На какую общую сумму продали только соков? (Функция СУММЕСЛИ)
- Какова средняя цена одного товара в чеке за эту неделю? (Функция СРЗНАЧ)
- Творческое задание: Какой товар принес столовой больше всего денег (сумма всех цен конкретного товара)? Нужно найти этот товар и написать его название.

Методический акцент:

Для слабых учеников важно разделить задачу: сначала мы фильтруем («где Товар = "Булочка"»), а потом считаем (СЧЁТЕСЛИ).

Учитель должен показать разницу между СЧЁТ (считает количество чисел) и СЧЁТЕСЛИ (считает количество ячеек, подходящих под условие, включая текст).

Практикум 2. «Мобильный гейминг: Статистика игрока» (Уровень: Средний / 7–8 классы)

Контекст: Работа с числовыми диапазонами и несколькими условиями. Контекст видеоигр сильно повышает мотивацию.

Исходные данные (фрагмент таблицы результатов матчей):

A	B	C	D
Никнейм	**Результат (убийства)**	**Время матча (мин)**	**Победа/ Поражение**
Gamer_1	12	15	Победа
Prosto_Ivan	3	22	Поражение
Sniper_OK	25	18	Победа
Gamer_1	8	25	Поражение
...	...	...	...
(Около 100 строк симулированных данных)			

Задание для учеников: Игрок хочет улучшить свою статистику. Помогите ему проанализировать базу из 100 игр:

- Посчитайте, сколько раз игрок с ником Gamer_1 одержал победу. (Подсказка: здесь понадобится суммирование по двум условиям - можно использовать вложенные знания или функцию СЧЁТЕСЛИМН, если она доступна в программе).
- Найдите среднее время матча, который закончился поражением. (Фильтрация по столбцу D, расчет СРЗНАЧ).
- Вычислите средний результат (количество убийств) игрока во всех матчах, которые длились меньше 20 минут. (Использование знаков сравнения < внутри функции СРЗНАЧЕСЛИ или СУММЕСЛИ / СЧЁТЕСЛИ для нахождения среднего арифметического вручную).
- Определите «Самый эффективный день»: создайте сводную мини-таблицу, которая покажет общее количество побед за каждый день недели (если добавлен столбец с датой).

Методический акцент:

Разбор типичной ошибки: ученики часто пытаются выделить диапазон условия мышкой вместе с заголовком столбца. Важно приучить их всегда указывать только чистые данные (например, C2:C101).

Обсуждение: почему среднее значение убийств может быть обманчивым, если есть один матч с результатом «0»?

Практикум 3. «Экологический мониторинг города» (Уровень: Продвинутый / 8–9 классы)

Контекст: Межпредметность (информатика + география/биология). Данные выглядят как настоящий научный массив.

Исходные данные: Таблица замеров качества воздуха в разных районах города за месяц.

Столбцы: `ID датчика`, `Район` (Центральный, Северный, Южный), `Концентрация PM2.5`, `Уровень шума (дБ)`.

(В таблице 365 строк — по одному замеру в день от каждого из 5 датчиков).

Задание для учеников: Эко-служба готовит доклад мэру. Подготовьте аналитическую справку:

- В скольких случаях за год уровень шума в Центральном районе превышал допустимую норму в 55 дБ? (Функция СЧЁТЕСЛИМН с двумя условиями: Район="Центральный" И Шум > 55).
- Какая средняя концентрация пыли (PM2.5) наблюдается в самом тихом районе города (где средний шум минимален)? (Это задание требует предварительного анализа средних значений по районам).
- Рассчитайте «Индекс плохой экологии». Создайте новый столбец. Если пыль > 30 ИЛИ шум > 60, ставьте «1», иначе «0». Затем посчитайте сумму этого столбца функцией СУММ. (Здесь вводится логика работы с массивами через вспомогательные столбцы, так как сложные формулы массива могут быть сложны для восприятия слабыми учениками).
- Визуализация: Постройте комбинированную диаграмму, показывающую динамику самого грязного показателя в течение месяца.

Методический акцент:

Учитель объясняет важность чистоты данных: перед использованием СРЗНАЧ нужно убедиться, что в колонках нет текстовых пометок типа «датчик сломан», которые функция проигнорирует, но которые указывают на пропуски в мониторинге.

Аккуратность ведения таблицы: использование абсолютных ссылок (B1), если ученики начинают применять критерии, вынесенные в отдельные ячейки.

Общие рекомендации для учителя:

- Цветовое кодирование: Просите учеников цветом выделять «Ареал вопроса» (какие колонки нам нужны?) и «Критерий» (что именно мы ищем?). Это визуализирует структуру формулы еще до её написания.
- Режим формул: Используйте комбинацию клавиш Ctrl+~ (или просмотр формул в меню), чтобы класс мог сравнить свои решения и увидеть синтаксис друг друга.

Развитие навыков самоконтроля у учеников 5–9 классов требует ухода от привычной схемы «учитель - судья». В информатике, где результат работы часто материален (файл, сайт, программа), ученикам проще оценивать качество продукта по объективным критериям, чем ставить оценку за абстрактные знания.

Ниже представлены пять методических приемов для формирования культуры самооценки и взаимопроверки на практических занятиях.

1. Прием «Светофор готовности» (Мгновенная визуальная диагностика)

Этот прием используется непосредственно в процессе выполнения задания, а не после него. Он учит ребенка вовремя замечать ступор и просить о помощи осознанно.

Инструментарий: У каждого ученика или пары на краю стола лежат три карточки: зеленая, желтая, красная (или стикеры).

Механика:

Зеленый: «Я понимаю задание, двигаюсь вперед самостоятельно».

Желтый: «У меня возник вопрос, я застрял, но могу подождать, пока вы освободитесь».

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

Как это развивает самооценку: Ученик вынужден проанализировать свое состояние перед тем, как поднять карточку. Это заменяет привычное махание рукой с криком «А у меня ничего не работает!» на рефлексию: «Какой именно шаг вызвал сложность? Мой уровень тревоги сейчас "желтый"».

2. Прием «Чек-лист пирата» (Рубрикатор успеха вместо оценки)

Слабые ученики боятся двоек. Чек-лист переводит фокус с наказания («ты сделал мало») на достижение («сколько пунктов ты собрал»).

Инструментарий: Перед началом работы выдается распечатанный список критериев. Рядом с каждым пунктом есть пустой квадратик.

Пример чек-листа для задания «Создание презентации»:

[ ] На первом слайде есть заголовок и имя автора.

[ ] Текст всех слайдов читаемый (шрифт не меньше 24 пт).
[ ] Все картинки подписаны снизу мелким шрифтом.
[ ] Настроена анимация смены слайдов (не вручную).
[ ] Файл сохранен в папке 7_Б_Фамилия под именем Проект.pptx.

Взаимопроверка: Ученики меняются местами. Задача проверяющего - не критиковать дизайн, а просто поставить галочки в пустых квадратах. Если галочки нет, он должен написать комментарий: «Здесь нет подписи к картинке льва». Оценка выставляется по количеству выполненных условий.

3. Прием «Реверсивное ревью» (Проверка чужого кода/проекта по ролям)

Чтобы оценить свою работу, нужно понять критерии качества. Проще всего научиться этому, оценивая другого.

Механика: Работа проводится в парах сменного состава.

Проверяющий получает строгую роль:

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

«Дизайнер»: Оценивает только внешний вид. Соответствие цветов, выравнивание элементов, отсутствие «поехавшей» верстки.

«Редактор»: Ищет только опечатки и грамматические ошибки в интерфейсе программы.

Методический нюанс: Запретите использовать фразы «Мне нравится» или «Это некрасиво». Разрешены только формулировки: «Нарушено правило №2», «Функция X выдает ошибку при значении Y». Это учит давать конструктивную, деперсонализированную обратную связь.

4. Прием «Дневник одной ошибки» (Анализ сбоев)

На уроках информатики ошибки неизбежны. Вместо того чтобы просто исправить баг за ученика, заставьте его отрефлексировать причину сбоя.

Инструментарий: Тетрадь или электронный документ «Лог ошибок».

Алгоритм заполнения после получения помощи от учителя:

Что пошло не так? (Описание: «Программа закрылась с ошибкой»).

Где это произошло? (Скриншот или название строки/команды).

Почему это случилось? (Гипотеза: «Я забыл сохранить файл перед запуском» или «Неправильно написал английское слово Input»).

Как я пойму, что ошибка вернулась? (Признак: «Появится красное окно с таким-то текстом»).

Результат: К концу года у слабого ученика формируется база знаний собственных типичных ошибок, и он начинает ловить их еще до нажатия клавиши Enter.

5. Прием «Галерея версий» (Сравнение промежуточных итогов)

Ученики часто думают, что работа должна получиться идеальной с первого раза. Демонстрация эволюции проекта снимает страх чистого листа.

Механика: Учитель просит сохранять проект каждые 10 минут под разными именами: Игра_v1, Игра_v2, Игра_Final.

Самооценка: В конце урока открывается папка со всеми версиями. Ученик отвечает письменно на вопросы:

Чем версия 2 отличается от версии 1?

Какое изменение сделало мой проект лучше всего?

Если бы времени было больше, какой элемент я бы улучшил в версии Final?

Взаимопроверка: Сосед смотрит только на Final и на v1. Его задача - найти одно самое удачное улучшение и похвалить за него одноклассника, а также указать на одну деталь, которая «заблудилась» по пути от первой версии к финальной.

Общие правила внедрения:

Для успешной работы этих приемов необходимо установить два жестких правила класса:

«Ошибаться можно»: Ошибка в практической работе - это повод заполнить «Дневник ошибки», а не получить плохую оценку.

«Сначала ищи сам, потом спрашивай соседа, потом поднимай красный светофор»: Это выстраивает иерархию самостоятельности и снижает нагрузку на учителя.

3.2.4. Методический анализ выполнения заданий обучающимися с высоким уровнем подготовки (в группе получивших отметку «5»), включая методические рекомендации для учителей

3.2.4.1. Описание предметной подготовки участников ОГЭ 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 группы «5» демонстрируют высокий уровень освоения всех содержательных линий курса информатики. Лучшие результаты: задание 1 (99,39%), задание 10 (98,79%), задание 5 (98,59%), задание 11 (98,59%). Практически все базовые и повышенные задания выполнены на уровне 93–99%. Задания высокого уровня сложности 14 (84,04%) и 15 (93,64%) также выполнены успешно.

- 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

Основные затруднения участников группы «5» связаны с заданием 16 – создание программ на универсальном языке программирования (36,26%). Даже наиболее подготовленные участники не в полной мере справляются с задачами, требующими полноценного программирования (анализ последовательностей чисел, работа с системами счисления в программном коде). Задание 13 (76,67%) и задание 14 (84,04%) также имеют резервы для повышения.

Таблица 2-6

№ п/п	Темы программы и умения	Класс(-ы) изучения в соответствии с ФОП
1.	Создание презентаций	7 класс
2.	Электронные таблицы	9 класс
3.	Программирование для исполнителя	8 класс

- Реалистичные «зоны роста» в части предметной подготовки для получения максимального количества первичных баллов.

Для получения максимального количества первичных баллов необходимо: повысить результативность выполнения задания 16 до 60–70% путём систематического решения задач на обработку последовательностей чисел с проверкой условий в различных системах счисления; достичь 100%-го выполнения практических заданий 13, 14 и 15. Рекомендуется участие в олимпиадах и конкурсах по программированию.

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

Общие рекомендации:

1) предлагать задачи повышенной сложности по программированию, включающие работу с различными системами счисления, обработку строк и числовых последовательностей;

2) организовать проектную деятельность с использованием электронных таблиц и средств визуализации данных;

3) привлекать к участию в олимпиадах и хакатонах по информатике;

4) практиковать решение задач с открытым ответом и самопроверкой;

5) развивать навыки оптимизации программного кода.

Для учеников 5–9 классов задачи повышенной сложности должны сочетать математическую логику с манипуляциями над текстом. Это развивает алгоритмическое мышление лучше, чем простое заучивание синтаксиса.

Ниже приведены комплексные задачи, каждая из которых требует использования циклов, условий и преобразования типов данных.

Задача 1. «Кибер-диспетчер: Декодирование позывных» (Комбинация систем счисления и строк)

Сюжет: Разведцентр перехватил сообщение от шпиона. Сообщение выглядит как бессмысленный набор букв и цифр, но на самом деле это координаты в разных системах счисления, скрытые внутри слов.

Условие: Дана строка S. Программа должна найти в ней все числа (последовательности цифр) и выполнить следующие действия:

Если число стоит сразу после гласной буквы (a, e, i, o, u, y - для упрощения), перевести его из двоичной системы в десятичную.

Если число стоит сразу после согласной буквы, перевести его из восьмеричной системы в десятичную.

Если число стоит в начале строки или после другого числа/знака препинания, оставить его без изменений (десятичным).

Вывести итоговую строку, где все найденные числа заменены результатами их перевода.

Пример ввода: `Agent_A101_code_77_status`

- `A` (гласная) + `101` -> $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 5$.
- `_` (не буква) + `77` -> оставляем как есть.

Пример вывода: `Agent_A5_code_77_status`

На что обратить внимание ученика:

- Нужно научиться отделять цифры от букв (метод .isdigit() в Python).

- Для перевода использовать функцию int(число, основание), предварительно собрав последовательность цифр в отдельную переменную-буфер.

- Важно правильно обработать границы чисел (когда буфер цифр заканчивается и нужно решить, переводить его или нет).

Задача 2. «Палиндром-термометр» (Сложная логика, системы счисления и симметрия)

Сюжет: В системе управления климатом произошел сбой. Данные о температуре приходят в зашифрованном виде. Нужно найти «особые» моменты времени.

Условие: Вводится диапазон лет (например, от 1950 до 2050). Год считается «критическим», если выполняется цепочка условий:

- Переведите год из десятичной системы в шестнадцатеричную.

- Запишите результат БУКВАМИ ЗАГЛАВНЫМИ (например, год 2024 это 7E8).

- Проверьте, является ли эта строка палиндромом (читается одинаково слева направо и справа налево). Буквы участвуют в проверке наравне с цифрами.

- Если строка палиндром, вычислите сумму всех её цифр (буквы A-F считать как 10-15).

- Если полученная сумма делится на семь, выведите этот год и его шестнадцатеричное представление.

Примеры для проверки:

Год 4113:

В 16-й системе: 1011.

Строка "1011" — не палиндром (1011 ≠ 1101). Не подходит.

Год 43981:
В 16-й системе: ABCD.
Не палиндром.
Год 25871:
В 16-й системе: 650F. Не палиндром.
(Поиск подходящего года можно предложить ученикам как часть исследования).
На что обратить внимание ученика:
- Функция hex() возвращает строку с префиксом 0x, который нужно отрезать через срез [2:].
- Чтобы проверить палиндром, проще всего сравнить строку с её перевернутым вариантом (s == s[::-1] в Python).
При подсчете суммы важно различать символы: если символ 'A', прибавить 10, если '5', прибавить 5. Для этого используется проверка if char.isalpha().

Задача 3. «Архивариус: Поиск скрытых дат» (Работа со строками и календарная логика)
Сюжет: В старинной книге страницы пронумерованы необычным образом — римскими цифрами. Из-за ветхости книги некоторые символы стерлись или были заменены на похожие латинские буквы. Нужно найти все возможные реальные даты в этом тексте.
Условие: Дана длинная строка S, состоящая из заглавных латинских букв. Программа должна проанализировать строку и найти в ней все подстроки длиной ровно 6 символов, которые могут представлять собой дату в формате ДДММГГ (две цифры дня, две цифры месяца, две цифры года), если прочитать их как римские числа.
Программа перебирает все возможные последовательности из 6 букв в строке.
Каждая последовательность разбивается на три пары: первые две буквы - день, вторые - месяц, третьи - год.
Напишите функцию, которая переводит пару "римских" букв в число. Для упрощения используйте значения: I=1, V=5, X=10, L=50.
Примечание: Чтобы задача была решаемой для данного возраста, можно игнорировать правило вычитания (где IV = 4) и просто суммировать значения символов (VI = 5+1=6, а IV = 1+5=6), либо жестко задать маппинг всех пар (II=2, IX=9 итд).
Проверьте полученную тройку чисел (день, месяц, год) на валидность:
Месяц должен быть от 1 до 12.
День должен соответствовать месяцу (февраль - до 29 дней, остальные месяцы - 30 или 31). Для простоты примем, что в феврале всегда 29 дней (високосный), чтобы не усложнять код проверкой делимости года на 4.
Если дата корректна, выведите её в формате DD.MM.YY -> Число_дня.Число_месяца.Число_года.
Пример ввода: ABXVIXLMMCD
Подстрока VIXLMM: День=VI(6), Месяц=XL(не может быть 90), Год=MM(2000) - Ошибка.
Подстрока IXLMMC: День=IX(9), Месяц=LM(ошибка), ...
Подстрока XIIVII: День=XII(12), Месяц=VII(7), Год=II(2) - Корректно! Дата: 12.07.02.
На что обратить внимание ученика:

- Использование вложенных циклов или срезов строки (s[i:i+6]) для сканирования текста.
- Создание вспомогательной функции roman_pair_to_int(pair), чтобы основной код оставался чистым.
- Работа со структурами данных: использование словаря (ассоциативного массива) для хранения значений букв { 'I': 1, 'V': 5, ... }.

Задача 4. «Код Да Винчи: Квадрат Полибия и простые числа» (Шифрование, матрицы и числовые фильтры)

Сюжет: Вы нашли записку, где каждая буква заменена на двузначное число (номер строки и номер столбца в таблице). Но шпионы передавали только те сообщения, сумма цифр которых является простым числом.

Условие: Используется русский алфавит (33 буквы), вписанный в квадрат 6 * 6 (буквы Ё нет или она объединена с Е; всего 36 клеток, лишние клетки пустуют или заполнены знаками препинания). Координаты начинаются с 11 (1-я строка, 1-й столбец). Алфавит заполняется построчно:

Строка 1: А(11), Б(12), В(13), Г(14), Д(15), Е(16)

Строка 2: Ж(21), З(22)...

Программа получает зашифрованную строку из чисел (например, "1122334416").

Разбейте входную строку на блоки по 2 символа. Каждый блок - это координата буквы.

Расшифруйте сообщение, превратив координаты обратно в текст.

Фильтрация: Для каждого блока-координаты вычислите сумму его цифр.

Пример: Блок 16 → 1 + 6 = 7.

Проверьте, является ли эта сумма простым числом (делится только на 1 и само себя). Простые числа до 18 (макс. сумма 6+6): 2, 3, 5, 7, 11, 13, 17.

Соберите итоговую фразу ТОЛЬКО из тех букв, чьи координаты дали простую сумму цифр.

Пример работы:

Ввод: `15223116`

1. Разбивка: `[15, 22, 31, 16]`.
2. Буквы без фильтрации: `Д, З, В, Е`.
3. Проверка координат:
   - `15` : $1 + 5 = 6$ (составное) $\rightarrow$ пропускаем.
   - `22` : $2 + 2 = 4$ (составное) $\rightarrow$ пропускаем.
   - `31` : $3 + 1 = 4$ (составное) $\rightarrow$ пропускаем.
   - `16` : $1 + 6 = 7$ (**простое**) $\rightarrow$ берем!
4. Вывод: `Е`.

На что обратить внимание ученика:

- Как превратить строку "1522..." в список чисел? Здесь отлично подойдет понимание того, что строку можно обрабатывать через шаг в цикле: for i in range(0, len(s), 2).

- Реализация проверки на простое число: достаточно проверить делимость числа на все числа от 2 до корня из этого числа (или просто перебором, так как суммы маленькие).

- Формирование новой строки путем добавления (+=) только пройденных проверку символов.

Эти задачи хороши тем, что имеют высокую реиграбельность: ученики могут сами менять ключи шифрования, правила перевода систем счисления или границы поиска дат, превращая задачу в собственный мини-проект.

Для сильных учеников стандартные задачи на написание кода или работу в офисных приложениях быстро становятся рутиной. Эффективные задания для этой категории должны содержать элемент неопределенности, требовать самостоятельного поиска информации и приводить к созданию уникального продукта.

Ниже представлены пять примеров заданий творческого, поискового и межпредметного характера.

1. Творческое задание: «Генеративное искусство ASCII» (Алгоритмика + Дизайн)

Суть: Создание программы, которая превращает любое изображение или текстовую фразу в художественный объект из символов псевдографики.

Задача: Написать программу (на Python/C#/Pascal), которая берет входной параметр (например, слово "ИНФОРМАТИКА") и визуализирует его не обычным шрифтом, а плотностью символов.

Механика: Ученик должен сам составить таблицу плотности символов (например, @ — самый темный/символ занимает много места, . - самый светлый). Программа должна выводить буквы так, чтобы их контур формировался за счет концентрации «плотных» знаков внутри силуэта символа.

Усложнение: Реализовать перевод реальной фотографии в ASCII-арт. Для этого нужно написать алгоритм, который уменьшает разрешение картинки до размеров терминала, считывает яркость каждого пикселя и заменяет его соответствующим символом из таблицы плотности.

Продукт: Интерактивная галерея портретов одноклассников в стиле киберпанк.

2. Поисковое задание: «Цифровая археология данных» (Работа с данными + Фактчекинг)

Суть: Восстановление истории по обрывочным цифровым следам.

Задача: Учитель предоставляет ученикам поврежденный датасет (CSV-файл или базу данных SQL) объемом от 500 строк. В файле отсутствуют названия столбцов, часть значений заменена на NULL, даты записаны вперемешку в форматах ДД.ММ.ГГГГ и MM/DD/YYYY, а числовые данные содержат мусор (текст вместо чисел).

Механика:

Используя инструменты очистки данных (встроенные фильтры Excel, скрипты на Python Pandas или SQL-запросы), ученик должен определить тематику данных (это могут быть логи сервера, метеорологические сводки XIX века или база продаж книжного магазина).

Найти минимум три аномалии («выброса»), которые выглядят как ошибки ввода, но при проверке оказываются исторически достоверными фактами.

Составить отчет о том, какие алгоритмы были применены для приведения данных к единому виду.

Продукт: Очищенная таблица и аналитическая справка с графиком распределения данных.

3. Межпредметное задание: «Биоинформатика: Алгоритм сжатия ДНК» (Информатика + Биология)

Суть: Применение классических структур данных для решения задач современной науки.

Задача: Генетический код состоит всего из четырех букв (A, T, G, C). Текстовый файл с геномом бактерии может весить мегабайты. Нужно разработать максимально эффективный способ хранения такой последовательности.

Механика:

Поскольку вариантов букв всего 4, их можно закодировать двумя битами (00, 01, 10, 11).

Сильный ученик должен реализовать это программно: превратить строку текста "ATGC..." в непрерывную битовую последовательность и упаковать её в байты.

Необходимо рассчитать коэффициент сжатия и сравнить свой результат со стандартным архиватором ZIP.

Усложнение: Реализовать алгоритм поиска паттернов. Программа должна находить внутри длинной цепочки заданный фрагмент (например, маркер вируса) быстрее, чем простым перебором (используя хэширование или деревья суффиксов).

Продукт: Утилита-компрессор биологических данных и расчет эффективности памяти.

4. Творческое задание: «Кибернетический оркестр» (Программирование + Музыка)

Суть: Создание алгоритмической музыки (процедурная генерация мелодий).

Задача: Написать программу, которая генерирует гармоничную мелодию, используя правила музыкальной теории, а не случайные числа.

Механика:

Ученик изучает базовые интервалы (квинты, терции) и законы построения ладов (мажор/минор).

Программа использует Марковские цепи: вероятность появления следующей ноты зависит от предыдущей (например, после ноты "До" чаще идет "Соль", реже - "Си-диез").

Звук выводится через системную MIDI-библиотеку или API синтезатора.

Усложнение: Добавить анализ настроения вводимого текста. Если пользователь пишет грустный стих, программа автоматически выбирает минорный лад и медленный темп; если веселый - мажорный.

Продукт: Веб-сервис или десктоп-приложение, создающее уникальные рингтоны.

5. Поисково-межпредметное задание: «Инфографика скрытых связей» (Анализ графов + История/Литература)

Суть: Визуализация сложных социальных или сюжетных сетей.

Задача: Выбрать сложный литературный произведение (например, «Война и мир», «Преступление и наказание») или исторический период (Династия Рюриковичей). Создать интерактивную карту взаимодействий.

Механика:

Сбор данных: ученики сами выписывают из книги или хроник факты взаимодействия персонажей/событий (Кто с кем встречался? Кто кому писал письма?).

Построение графа: использование библиотек типа networkx (Python) или специализированных редакторов диаграмм.

Анализ метрик: вычисление «центральности». Кто является самым важным узлом в сюжете не по количеству страниц, а по количеству связей с другими героями?

Результат: Интерактивная карта (HTML-страница), где размер узла соответствует значимости персонажа, а цвет ребра - типу связи (конфликт/дружба). При наведении мыши отображается цитата из первоисточника, подтверждающая эту связь.

Продукт: Цифровой исследовательский атлас произведения.

Эти задания требуют от учителя роли наставника-консультанта, а не проверяющего, так как единого правильного пути решения здесь не существует.

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 ОГЭ

На основе выявленных сложных для обучающихся тем и типичных ошибок, а также предметных и методических дефицитов учителей информатики можно сформировать актуальную повестку для заседаний школьного методического объединения (ШМО) учителей информатики.

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

Блок 1. Методический практикум: «Преодоление предметных дефицитов»

Цель: Разбор конкретных тем, вызывающих наибольшие трудности у учеников разного уровня.

- Тема: «Системы счисления без паники».

Обсуждение: Как уйти от заучивания степеней двойки? Обмен наработками по приему «Весы со степенями» и заданиям с римскими цифрами в коде.

Практика: Коллеги решают задачу «Кибер-диспетчер» друг друга, оценивая понятность инструкции для слабого ученика.

- Тема: «Таблица - лучший дебаггер: обучение трассировке через таблицы значений».

Обсуждение: Приемы формирования самоконтроля. Как заставить ученика заполнять таблицу переменных, не заглядывая в ответ?

Обмен опытом: Демонстрация удачных шаблонов таблиц для заданий №6 (Робот-обменник) и №3 (Умный счетчик).

- Тема: «Поиск информации как алгоритм: от "скачал реферат" к фактчекингу».

Обсуждение: Переход от обучения слепому копированию к навыкам верификации данных.

Разбор кейсов: Задания типа «Цифровой детектив» и «Битва титанов» - как оценивать процесс поиска, а не только финальный текст.

Блок 2. Инструментарий современного урока: «ИИ и автоматизация рутины»

Цель: Снизить нагрузку учителя и повысить вовлеченность учеников за счет технологий.

- Тема: «Учитель как промпт-инженер: использование нейросетей для генерации дидактики».

Обсуждение: Создание банка промптов. Как одной командой генерировать 5 вариантов контрольной работы по теме «Алгоритмизация», 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 задач (например, обложка для квеста).

- Тема: «Автоматическая проверка работ: спасение утопающих».

Обсуждение: Использование скриптов для проверки практических работ. Например, скрипт, который сам запускает программы учеников, сравнивает вывод с эталоном и ставит предварительную оценку.

Экономия времени: Сколько часов в неделю экономит автоматизированная проверка трассировочных таблиц или простых программ.

Блок 3. Работа с гетерогенными группами (разноуровневое обучение)

Цель: Найти способы удержать внимание сильных учеников, не потеряв при этом слабых.

- Тема: «Дифференциация внутри одного задания: концепция "Лестницы"».

Обсуждение: Как одно задание (например, обработка строк) может быть базовым для одних («найди количество гласных») и исследовательским для других («декодируй сообщение со сдвигом Цезаря»).

Обмен банком задач: Презентация тех самых 5–7 сложных межпредметных задач (генеративное искусство, биоинформатика), которые выдаются сильным ученикам после выполнения основной части работы.

- Тема: «Парное программирование: кто везет, а кто учится?»

Обсуждение: Роли «Штурман» и «Архивариус» (или «Детектив» и «Оперативник»). Как распределять роли так, чтобы сильный ученик не делал все за двоих, а слабый не просто смотрел в монитор.

Методика: Опыт внедрения взаимной оценки по чек-листам перед сдачей работы учителю.

Блок 4. Трансформация системы оценивания деятельности

Цель: Адаптировать систему оценки под требования времени.

- Тема: «От двоек за синтаксис к оценке логики: новые критерии практической работы».

Обсуждение: Если нейросеть пишет код за секунду, что мы оцениваем? Ошибку компиляции или неверную декомпозицию задачи?

Критерии: Введение весовых коэффициентов: 20% - синтаксис, 80% - алгоритм, читаемость кода и комментарии.

- Тема: «Межпредметные проекты: как «продавать» информатику коллегам».

Обсуждение: 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.

Целесообразно использовать интерактивные форматы проведения ШМО:

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

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

- 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

Представим направления для ИРО, разделенные по форматам реализации и целевой аудитории.

Блок 1. Долгосрочные модульные курсы
Целевая аудитория: Учителя со стажем до 5 лет; педагоги, работающие в классах с выраженной предметной дифференциацией.
- Название курса: «Алгоритмика для "гуманитариев": методика объяснения сложного через визуализацию».
Цель: Устранение барьера при изучении базовых алгоритмов у слабых учеников 5–9 классов.
Модули:
1) Трассировка как инструмент самоконтроля: обучение работе с таблицами значений переменных («Дневник одной ошибки», «Светофор готовности»).
2) Системы счисления без страха: игровые методики (шестнадцатеричный шифр, квадрат Полибия), перевод абстрактных понятий в осязаемые действия.
3) Обучение чтению кода vs написанию кода: методы взаимопроверки по чек-листам («Чек-лист пирата») и парное ревью.
- Название курса: «Межпредметная информатика: от биоинформатики до генеративного искусства».
Цель: Подготовка педагогов к работе с высокомотивированными детьми и реализации индивидуальных проектов.
Модули:
1) Работа со строками и регулярными выражениями на примере криптографии и обработки ДНК-последовательностей.
2) Основы процедурной генерации контента (ASCII-арт, фракталы) для уроков дизайна и математики.
3) Анализ данных: использование Python/Excel для обработки больших массивов информации (от школьных опросов до экологических данных).
Название курса: «Алгоритмическое мышление без страха: методика обучения слабых учеников программированию».
Цель: Снятие барьера перед трассировкой и переменными.
Модули:
1) Декомпозиция алгоритмов: использование таблиц значений как основного инструмента самоконтроля.
2) Игровые форматы обучения (Роботы, Черепашки, Магические коробки) для формирования базовых понятий.
3) Обучение чтению кода vs написанию кода: методики взаимопроверки по чек-листам.

4) Работа с типичными ошибками перехода (ошибки «на единицу» в циклах, путаница в системах счисления).

Блок 2. Краткосрочные интенсивы и практикумы

Целевая аудитория: Все категории учителей, нуждающиеся в обновлении инструментария.

- Формат-практикум: «Нейросети в роли учителя-программиста».

Фокус: Использование ИИ для создания дидактического материала и автоматизации проверки.

Содержание: Как генерировать уникальные варианты задач по системам счисления или обработке строк с помощью промптов. Создание скриптов-декораторов для автоматической проверки практических работ учеников (Unit-тестирование для школы).

- Лаборатория: «Ошибкоустойчивое программирование».

Фокус: Разбор сложных олимпиадных конструкций из прошлых запросов.

Содержание: Совместный разбор и реализация заданий «Квадрат Полибия», «Генератор паролей Фибоначчи» и «Архивариус». Учителя сами выступают в роли учеников, чтобы понять когнитивную нагрузку таких задач.

- Практикум: «Робототехника и "железо" для начинающих».

Фокус: Переход от виртуального программирования (экран) к реальному (контроллеры Arduino/micro:bit).

Содержание: Сборка простых устройств, демонстрирующих работу циклов и условий в физическом мире (например, автоматический полив растений или кодовый замок).

- Формат-практикум: «Межпредметные исследования на уроках математики и информатики».

Фокус: Реализация идей биоинформатики, фрактальной геометрии и теории игр.

Содержание: Участники под руководством эксперта пишут скрипты для генерации ASCII-арта, анализируют реальные наборы данных (например, экологический мониторинг) и создают игровые модели баланса ресурсов.

- Лаборатория: «Нейросети в работе учителя-предметника».

Фокус: Трансформация ВСОКО и подготовка дидактики.

Содержание: Как генерировать уникальные варианты контрольных работ разного уровня сложности одним промптом. Использование ИИ для создания качественных визуализаций тел вращения и графиков функций для раздаточного материала.

Блок 3. Форматы наставничества и тьюторского сопровождения

Целевая аудитория: Молодые специалисты; учителя сельских школ.

- «Предметная клиника кода»: регулярные сессии, где учитель приносит фрагмент неработающего кода ученика или сложный проект. Группа экспертов разбирает логические ошибки публично, обучая методике задавания правильных наводящих вопросов вместо прямого исправления.

- Серия консультаций «От идеи к проекту»: помощь учителям в оформлении исследовательских работ школьников на конференции. Эксперт помогает перевести бытовую идею («почему мой телефон греется?») в формат измеримого ИТ-проекта (замеры температуры процессора, построение графиков зависимости нагрузки от времени).

- Методический аудит рабочих программ: корректировка КТП с учетом введения тем по искусственному интеллекту и информационной безопасности, которые часто игнорируются из-за нехватки компетенций педагога.

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

Для масштабирования опыта организациям рекомендуется создать следующие продукты:

- Банк интерактивных тренажеров «Трасса-ПРО». Веб-сервис, позволяющий учителю вставлять любой фрагмент кода, а ученику - пошагово заполнять таблицу значений переменных прямо в браузере с автоматической проверкой каждого шага заполнения таблицы.
- Конструктор межпредметных кейсов. Библиотека готовых сценариев уроков на стыке дисциплин: «Информатика + Биология (ДНК)», «Информатика + Литература (анализ текстов классиков)», «Информатика + Экономика (базовая аналитика продаж)».
- Библиотека эталонных решений повышенной сложности. Видеоразборы задач творческого характера (генерация музыки, ASCII-арт) с подробным комментарием учителя: почему выбран именно такой тип данных, какие есть подводные камни производительности.

Представленные методические мероприятия целесообразно организовать на базе стажировочных площадок - образовательных организаций, демонстрирующих высокие результаты ГИА.

СОСТАВИТЕЛИ ОТЧЕТА по учебному предмету: **ИНФОРМАТИКА**

Специалисты, привлекаемые к анализу результатов ОГЭ 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	Государственное образовательное учреждение Республики Коми «Физико-математический лицей-интернат», учитель информатики, председатель республиканской предметной комиссии по проверке экзаменационных работ при проведении государственной итоговой аттестации по образовательным программам основного общего образования по информатике
Габова Марина Анатольевна	ГОУДПО «Коми республиканский институт развития образования», проректор по научно-методической работе, кандидат педагогических наук, доцент

Специалисты, привлекаемые к подготовке методических рекомендаций на основе результатов ОГЭ 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.)*
Габова Марина Анатольевна	ГОУДПО «Коми республиканский институт развития образования», проректор по научно-методической работе, кандидат педагогических наук, доцент

Ответственный специалист в Республике Коми по вопросам организации проведения анализа результатов ОГЭ 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