



ФЕДЕРАЛЬНЫЙ ИНСТИТУТ ПЕДАГОГИЧЕСКИХ ИЗМЕРЕНИЙ

**В.Р. Лещинер**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО  
НЕКОТОРЫМ АСПЕКТАМ  
СОВЕРШЕНСТВОВАНИЯ  
ПРЕПОДАВАНИЯ  
ИНФОРМАТИКИ И ИКТ**

*(на основе анализа типичных затруднений выпускников при  
выполнении заданий ЕГЭ)*

Москва, 2014

Целью Единого государственного экзамена является установление уровня освоения выпускниками Федерального компонента государственного образовательного стандарта среднего (полного) общего образования по информатике и ИКТ. Используемые при этом контрольные измерительные материалы позволяют соотнести результаты, показанные отдельными экзаменуемыми, путем выставления за работу количественной оценки по стобалльной шкале. Таким образом, становится возможным использовать результаты ЕГЭ для конкурсного отбора абитуриентов вузов.

Результаты единого государственного экзамена по информатике и ИКТ признаются образовательными организациями среднего профессионального и высшего образования как результаты вступительных испытаний по информатике и ИКТ. ЕГЭ по информатике и ИКТ входит в перечень вступительных испытаний для специальностей высшего образования, не только непосредственно связанных с ИКТ и вычислительной техникой, но и многих общеинженерных, технологических специальностей, а также для специальностей картографического и гидрометеорологического направлений, физико-математических специальностей классических и педагогических университетов.

Единый государственный экзамен проверяет знания и умения выпускников по результатам обучения в средней школе. Однако, полученное в рамках основной школы образование по предмету является фундаментом для обучения в средней школе: без овладения содержанием и достижения требований, предусмотренных стандартом основной школы, невозможно достижение требований стандарта среднего (полного) общего образования.

Структура и объем учебного плана по информатике в образовательных организациях разных типов и видов сильно варьирует: от 240 часов в старших классах информационно-технологического профиля до 70 часов базового курса в классах гуманитарных профилей. В этой связи КИМ ЕГЭ содержат задания, рассчитанные как на выпускников профильных классов, так и на тех, кто прослушал только базовый курс для средней школы (то есть задания, проверяющие достижение требований как базового, так и профильного стандарта).

Каждое задание экзаменационной работы характеризуется одновременно проверяемым содержанием учебного предмета, требованием к подготовке выпускника, достижению которого проверяется с помощью этого задания, а также уровнем сложности и характером осуществляемой деятельности (воспроизведение знаний и умений, применение их в стандартной либо новой ситуации). Структура экзаменационной работы моделируется исходя из принципов всесторонней проверки: задания экзаменационной работы должны проверять все существенные элементы содержания предмета и достижение всех требований к уровню подготовки, которые возможно проверить в формате ЕГЭ.

Экзаменационная работа ЕГЭ 2014 г. по информатике и ИКТ содержала 32 задания и состояла из трёх частей. В каждой из частей были сгруппированы задания с одинаковым форматом ответа. В первой части работы (А) содержалось 13 заданий с выбором одного правильного ответа из четырех предложенных. Во второй части (В) были собраны 15 заданий с краткой формой ответа, подразумевающие самостоятельное формулирование и ввод ответа в виде последовательности символов. Третья часть (С) содержала 4 задания, требовавшие записать в произвольной форме развернутый ответ на специальном бланке для последующей проверки экспертами.

Экзамен проверяет знания и умения учащихся по десяти разделам курса информатики. При этом удельный вес разделов в экзамене различен и примерно соответствует значению соответствующего раздела в курсе. Наибольшее количество заданий в курсе приходится на разделы «Элементы теории алгоритмов» и

«Программирование», что связано с ведущей ролью вопросов алгоритмизации и программирования в курсе.

Содержание экзамена составлено таким образом, чтобы на результат не влияло существенным образом то, по какой программе или учебно-методическому комплексу велось преподавание в конкретной образовательной организации, какое программное обеспечение использовалось, хотя отсутствие тех или иных предусмотренных стандартом элементов содержания в конкретном учебном курсе будет, естественно, влиять на итоговую оценку подготовки выпускников. Также очевидно, что полностью исключить влияние компьютеризации учебного процесса в образовательном учреждении на результаты ЕГЭ по информатике и ИКТ невозможно, но содержание экзаменационной работы позволяло выпускникам, изучавшим информатику и ИКТ в «безмашинном» варианте, преодолеть минимальную границу и получить достаточный для поступления на непрофильную техническую специальность балл.

При разработке критериев оценивания заданий с развернутым ответом учитывается факт разнообразия учебных программ и изучаемых языков программирования. При выполнении заданий С2 и С4, предполагающих написание программы, экзаменующийся может использовать любой язык программирования, требуется только точно назвать версию языка. В связи с этим критерии оценивания пишутся в общей форме, и не зависят от выбранного языка программирования. Этот подход предъявляет высокие требования к квалификации экспертов, проверяющих задания: они должны быть готовы проверять задания, выполненные на различных языках программирования. Также во всех заданиях, содержащих фрагменты алгоритмов, записанных на формальном языке, они предъявляются на четырёх различных языках программирования: школьном Алгоритмическом языке, Си, Паскале и Бейсике.

Критерии оценивания заданий с развернутым ответом учитывают невозможность для экзаменуемых проверить решение в процессе отладки в среде программирования и поэтому толерантны к синтаксическим ошибкам, не искажающим замысел автора программы.

Экзамен проверял знания и умения выпускников с использованием заданий различного уровня сложности: базового (15 заданий), повышенного (13 заданий) и высокого (4 задания).

Из 15 заданий базового уровня по крайней мере 12 не вызывают затруднений у экзаменуемых. Задание А9, проверяющее умение кодировать информацию с использованием неравномерного кода, выполняется экзаменуемыми с результатом (59,6%), соответствующим нижней границе, установленной для заданий базового уровня. Содержание этого задания в 2014 г. было связано с определением минимального по длине двоичного кода, обеспечивающего однозначное декодирование. Опыт нескольких лет показывает, что экзаменуемые лучше справляются с заданиями на равномерное кодирование, чем с заданиями, связанными с использованием неравномерного по длине кода. При изучении основ информатики и ИКТ следует обратить внимание на раздел, связанный с кодированием, более подробно разбирать понятие префиксного кода, сопровождая это решением задач.

Определенные затруднения экзаменующихся вызвало задание В4, проверяющее на уровне применения в стандартной ситуации знания об алфавитном методе измерения количества информации. Результаты выполнения этого задания варьируют в диапазоне от 35 до 51%. Этого явно недостаточно для задания базового уровня, проверяющего знания по одной из ключевых тем школьного курса. Приведем анализ решения одной из типичных задач В4 из КИМ ЕГЭ 2014 г.

#### **Формулировка задания В4:**

Все 4-буквенные слова, составленные из букв К, Л, Р, Т, записаны в алфавитном порядке и пронумерованы.

Вот начало списка:

1. КККК
2. КККЛ
3. КККР
4. КККТ

.....

Запишите слово, которое стоит под номером **67**.

#### **Решение (1 способ):**

Нужно посчитать, сколько слов всего в этом списке (сколько 4-буквенных слов можно составить из 4 различных букв). Так как на каждой из 4 позиций может стоять любая из 4 букв, то количество слов в списке будет равно  $4^4 = 256$ .

При этом слова под номерами 1-64 будут начинаться на букву К, под номерами 65-128 будут начинаться на букву Л. Под номером 65 стоит слово ЛККК, а под номером 67 – ЛККР.

#### **Решение (2 способ):**

Обозначим К = 0, Л = 1, Р = 2, Т = 3. Тогда получается, что список состоит из чисел в системе счисления с основанием 4, количество значащих цифр в которых не превышает 4.

Таких чисел  $4^4 = 256$ . При этом число 0000 в списке пронумеровано 1, 0001 – 2, а под номером 67 будет стоять число, равное  $66_{10}$ , которое и нужно будет перевести в четверичную систему счисления.

Разложим 66 на степени четверки ( $4^3=64$ ,  $4^2=16$ ,  $4^1=4$ ,  $4^0=1$ ):  $66 = 1 \cdot 64 + 0 \cdot 16 + 0 \cdot 4 + 2 \cdot 1$

Выписав цифры, получим  $66_{10} = 1002_4$ . Вспомнив начальные обозначения, запишем искомое слово: ЛККР.

#### **Ответ: ЛККР**

Задание В6 проверяет владение экзаменуемыми понятием рекурсии и связанными с ним умениями и навыками. Оно в 2014 г. было выполнено плохо (средний процент выполнения 13,2%). Можно предположить, что это связано с изменением содержания данного задания в 2014 г. по сравнению с 2013 г. В 2014 г. задание представляло собой фрагмент рекурсивного алгоритма, содержащего оператор вывода (печати) внутри вызываемой рекурсивной функции. Вопрос формулировался просто: какое количество символов (чисел) будет напечатано, то есть сколько раз будет выполнена команда вывода. По идее, это задание решается методом формального исполнения (трассировки) алгоритма, то есть в результате репродуктивной деятельности, знакомой учащимся. Низкий показатель выполнения этого задания говорит о том, что понятие рекурсии многими учащимися в процессе обучения так и не было освоено.

Задания повышенного уровня (их в работе 13 из 32, и содержатся они во всех трех частях работы) проверяют содержание профильного уровня стандарта 2004 г. по информатике и ориентированы на оценку подготовки выпускников, изучавших предмет по углубленной программе. Правильное решение этих заданий позволяет экзаменуемому получить 37,5% максимального первичного балла. Из заданий повышенного уровня 5 заданий посвящены основаниям информатики (разделы «Системы счисления», «Основы логики», «Информация и её кодирование», «Моделирование»), 2 задания – информационно-коммуникационным технологиям (разделы «Архитектура компьютеров и компьютерных сетей», «Технологии поиска и хранения информации») и 6 заданий посвящены основам теории алгоритмов и

программированию. К темам, которые проверялись в заданиях повышенного уровня, относятся следующие: позиционные системы счисления; определение истинности логических выражений; информационный объем сообщения; определение скорости передачи информации при заданной пропускной способности канала; адресация в Интернете; поиск при помощи сложных запросов; подсчет числа путей в графе; анализ алгоритмов, в том числе алгоритмов обработки массивов и программ, содержащих вызов пользовательской функции, поиск ошибок в программе).

Из 13 заданий повышенного уровня 8 заданий (A10, A11, A12, A13, B9, B10, B11, B12) были выполнены экзаменуемыми с результатом, соответствующим уровню сложности задания (от 40 до 60%). Результаты выполнения задания повышенного уровня сложности с развернутым ответом C1 (средний результат 36,7%, разброс по вариантам от 27 до 46%) также можно считать удовлетворительными, несмотря на новый элемент, введенный в задание в 2014 г.

Задания C1 этого года требовали не только указать, каков будет результат работы программы при определенных входных значениях, но и привести пример такой входной последовательности, при которой программа, по условию содержащая ошибки, выдаст верный результат. Для того, чтобы найти такую последовательность, требуется понять, какие ошибки были допущены при написании программы, то есть сформулировать общее правило. Пример такого правила для одной из задач: «В конце работы программы значение переменной `maximum` всегда равно 999, независимо от того, какие числа были введены. Соответственно, программа будет работать верно, если в последовательности есть 999 и это максимальное число.» Также новым элементом в системе оценивания этого года было то, что оценка снижалась за найденные «ошибки» (такими не являвшиеся) сверх имевшихся в программе двух ошибочных строк. Нововведения этого года в системе оценивания немного снизили результат выполнения задания C1, но, тем не менее, целесообразно оставить их в заданиях такого рода, так как они повышают качество выполнения задания, отсеивая случайно угаданные ошибочные строки.

4 задания повышенного уровня сложности вызвали у экзаменуемых серьезные затруднения. В первую очередь, это задание B7, которое проверяет знание правил записи чисел в позиционных системах счисления с произвольным основанием. В 2014 г. были использованы принципиально новые формулировки задания, проверяющие глубокое понимание правил записи чисел в двоичной системе. Типичная формулировка этого задания выглядит следующим образом: «Сколько единиц в двоичной записи значения выражения  $4^{2015} - 2^{2014} + 3$  ?»

Для решения этой задачи экзаменуемый должен твердо знать, что число  $2^k$  ( $4^k$ ,  $8^k$  и т.д.) всегда записывается в двоичной системе как единица с соответствующим количеством нулей. Разность  $2^k - 2^n$  при  $k > n$  выглядит в двоичной системе как  $k-n$  единиц с последующими  $n$  нулями. Этот факт – наглядный результат выполнения в двоичной системе алгоритма вычитания «в столбик» (с займом единицы в старшем разряде). Убедиться в справедливости этого правила можно, записав в двоичной системе значение разности  $128-32$  ( $2^7-2^5$ ) =  $96 = 64+32 = 110000_2$

Таким образом, число  $4^{2015}$  записывается в двоичной системе как единица с последующими 4030 нулями. Разность  $4^{2015} - 2^{2014}$  выглядит как 2016 единиц и последующие 2014 нулей, а прибавление числа 3 ( $11_2$ ) к этой разности заменяет два последних нуля единицами (все по алгоритму сложения в столбик). Итак, в двоичной записи этого выражения 2018 единиц.

Может показаться, что это задание относится к категории «олимпиадных задач», так как ставит экзаменуемого в тупик невозможностью прямого решения. Однако, это не так. Олимпиадное задание отличается тем, что не содержит прямого указания на то, каким элементом знания (фактом, алгоритмом, правилом) следует воспользоваться для его решения. В данном случае экзаменуемый точно знает, что

ему следует применить общее правило записи чисел в позиционной системе счисления и хорошо знакомый с начальной школы алгоритм сложения и вычитания «столбиком». Задача легко решается в уме без громоздких вычислений.

Трудоемкостью отличается и задание повышенного уровня сложности В13 проверяющее умение анализировать результат исполнения алгоритма. Это задание было выполнено примерно 35% экзаменуемых. В формулировке 2014 г. задание проверяет умение построить дерево путей вычислений, а также, умение анализировать такое дерево, например, вычислять количество путей, удовлетворяющих заданным свойствам. Задание может выглядеть, например, так. Дано описание исполнителя, который работает с натуральными числами. Команды этого исполнителя преобразуют данное число в новое число, которое больше исходного. Программа для такого исполнителя – последовательность команд. Требуется подсчитать количество программ, которые переводят число  $a$  в число  $b$ .

#### **Формулировка задания В13:**

У исполнителя Увеличитель две команды, которым присвоены номера:

- 1. прибавь 1,**
- 2. умножь на 2.**

Первая из них увеличивает число на экране на 1, вторая – умножает его на 2.

Программа для Увеличителя – это последовательность команд. Сколько есть программ, которые число 3 преобразуют в число 23?

Задачу можно решать двумя способами. Первый способ – выписать все нужные программы, построить дерево программ. Второй – подсчитать число программ, не выписывая их явно, а написав формулу, которая позволяет найти количество программ получения данного числа, если уже известно количество программ для получения меньших чисел (при таком решении удобно заполнять таблицу).

Первый способ более нагляден, но пригоден только тогда, когда количество программ невелико. Кроме того, при таком способе решения даже сильные ученики могут допускать неточности при записи решений. Поэтому мы советуем разобрать с учениками, как первый, так и второй способы решения.

#### **Решение (1 способ):**

Будем строить дерево решений (см. рис.1). Ребра в нем будут помечены допустимыми командами, в каждом узле будет записано текущее число, т.е. число, полученное из числа 3 с помощью последовательности команд, которые соответствуют пути из корня в этот узел. В корне будет записано число 3. Нас интересует количество путей из корня в вершины, помеченные числом 23. Так как обе команды увеличивают исходное число, то вершины, соответствующие числам, большим, чем 23, рассматривать не нужно.

На рисунке линия (без стрелки), идущая влево, соответствует команде **1. прибавь 1**; линия (без стрелки), идущая вправо, соответствует команде **2. умножь на 2**.

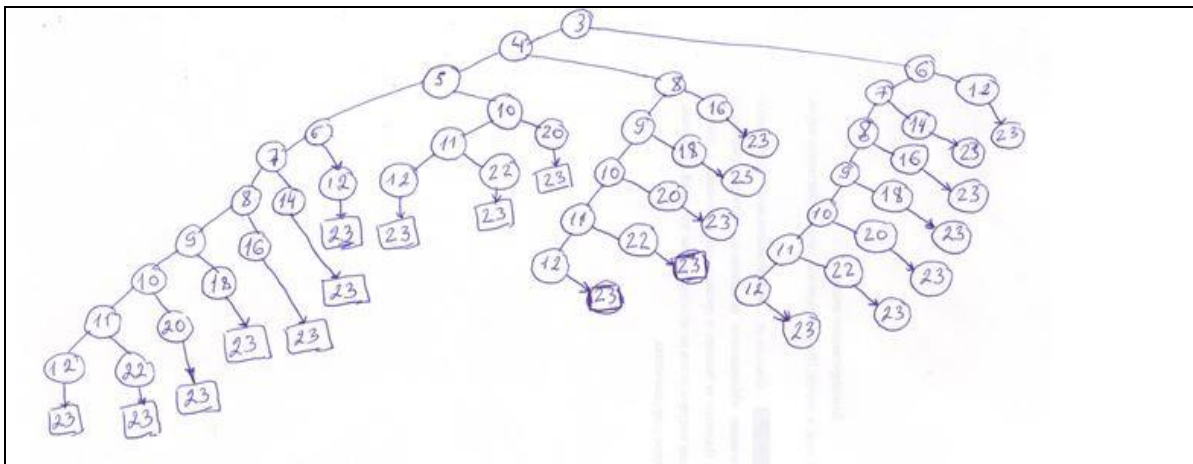


Рис.1. Дерево возможных путей вычислений.

Отметим, что если в вершине записано число  $n$  от 12 до 22, то из этой вершины есть ровно один путь в вершину с пометкой 23. Действительно, один такой путь всегда есть, он соответствует  $23-n$  ребрам с пометкой 1 (т.е. выполнению команды «прибавь 1»  $23-n$  раз). Если мы «свернем» с этого пути, т.е. находясь, в промежуточной точке выполним команду «умножь на 2», то мы получим число, большее, чем 23. Т.к. обе наши операции увеличивают исходное число, то далее мы не сможем получить число 23. Для краткости такой единственный путь от числа  $n$ , где  $9 \leq n < 32$  к числу 32 на рисунке изображен стрелкой.

Осталось пересчитать количество вершин с пометкой 23.

**Ответ: 22**

Заметим, что при решении построением дерева приходится делать лишнюю работу. Например, число 6 в дереве появляется дважды: один раз после выполнения команды «умножь на 2», другой – после троекратного выполнения команды «прибавь 1». Обработав его один раз (например, после умножения 3 на 2), можно подсчитать, что есть 7 способов получить число 23 из числа 6. Поэтому строить второй раз поддерево с корнем 6 не требуется. Второй способ решения (с помощью построения таблицы) и есть способ систематически избегать лишней работы.

Отметим, что, если бы вместо числа 23 взяли, скажем, число 50 (и, заведомо, если взять 100), то, пожалуй, задача станет неподъемной. А описанным ниже методом решения с помощью заполнения таблиц это сделать довольно легко.

**Решение (2 способ):**

Будем решать поставленную задачу последовательно для чисел 3, 4, ..., 23. То есть для каждого из чисел определим, сколько программ исполнителя существует для получения этого числа из числа 3. Количество программ, которые преобразуют число 3 в число  $n$ , будем обозначать через  $R(n)$ . Для удобства будем заполнять такую таблицу (см. рис. 2а). Для удобства таблица разбита на две части – для чисел от 3 до 13 и для чисел от 14 до 23.

Число $n$	3	4	5	6	7	8	9	10	11	12	13
Из каких чисел можно получить число $n$ за один шаг											
Количество программ, с помощью которых можно получить число $n$ из числа 3 (не обязательно за один шаг)											

Число	14	15	16	17	18	19	20	21	22	23
Из чего можно получить число $n$ за один шаг										
Количество программ, с помощью которых можно получить число $n$ из числа 3 (не обязательно за один шаг)										

Рис 2а.

В таблице средняя строка показывает, из каких чисел может быть получено данное число за одно действие. Таблицу удобно заполнять так. Двигаясь от меньших чисел к большим, то есть в порядке 3, 4, ..., 22, для каждого числа отмечаем, какие числа могут быть получены из этого числа. На рис. 2б показана верхняя часть таблицы после обработки чисел 3 и 4; на рис. 2в – эта же часть после того, как дополнительно обработано число 5.

На рисунке 2г средняя строка таблицы заполнена полностью. Если число не делится на два, то оно может быть получено только из предыдущего с помощью команды **прибавь 1**. Если число на 2 делится, то вариантов последней команды два: **прибавь 1** и **умножь на 2**,



Число $n$	3	4	5	6	7	8	9	10	11	12	13
Из каких чисел можно получить число $n$ за один шаг		3	4	3		4					
Количество программ, с помощью которых можно получить число $n$ из числа 3 (не обязательно за один шаг)											

Рис. 2б

Число $n$	3	4	5	6	7	8	9	10	11	12	13
Из каких чисел можно получить число $n$ за один шаг		3	4	3, 5		4		5			
Количество программ, с помощью которых можно получить число $n$ из числа 3 (не обязательно за один шаг)											

Рис. 2в

Теперь перейдем к заполнению нижней строки. Число 3 у нас уже есть, значит, его можно получить с помощью “пустой” программы. Любая непустая программа увеличит исходное число, т.е. даст число, больше 3. Значит,  $R(3) = 1$ . Для каждого следующего числа рассмотрим, из какого числа оно может быть получено за одну команду исполнителя.

Если число не делится на два, то оно может быть получено только из предыдущего с помощью команды **прибавь 1**. Значит, количество искомых программ для такого числа равно количеству программ для предыдущего числа:  $R(n) = R(n-1)$ . Если число на 2 делится, то вариантов последней команды два: **прибавь 1** и **умножь на 2**, тогда  $R(n) = R(n-1) + R(n/2)$ .

В общем виде это можно записать так:

$$R(n) = R(p_1) + R(p_2),$$

где  $p_1, p_2$  – числа, из которых число  $n$  можно получить за одно действие. Понятно, что если у числа  $n$ , не 2, а, один «предшественник», то формула будет выглядеть  $R(n) = R(p)$ .

Число $n$	3	4	5	6	7	8	9	10	11	12	13
Из каких чисел можно получить число $n$ за один шаг		3	4	3, 5	6	4, 7	8	5, 9	10	6, 11	12
Количество программ, с помощью которых можно получить число $n$ из числа 3 (не обязательно за один шаг)											

Число	14	15	16	17	18	19	20	21	22	23
Из чего можно получить число $n$ за один шаг	7, 13	14	8, 16	16	9, 17	18	10, 19	20	11, 21	22
Количество программ, с помощью которых можно получить число $n$ из числа 3 (не обязательно за один шаг)										

Рис 2г.

Пользуясь формулой сложения, двигаясь от меньших чисел к большим, можно заполнить нижнюю строку таблицы. Как сказано выше,  $R(3) = 1$ . Далее, по формуле сложения и используя уже заполненную среднюю строку, получаем:  $R(4) = R(3) = 1$ ;  $R(5) = R(4) = 1$ ;  $R(6) = R(3) + R(5) = 1+1 = 2$  и т.д. Полностью заполненная таблица приведена на рисунке 3.

Число $n$	3	4	5	6	7	8	9	10	11	12	13
Из каких чисел можно получить число $n$ за один шаг		3	4	3, 5	6	4, 7	8	5, 9	10	6, 11	12
Количество программ, с помощью которых можно получить число $n$ из числа 3 (не обязательно за один шаг)	1	1	1	$1+1 = 2$	2	$1+2 = 3$	3	$1+3 = 4$	4	$2+4 = 6$	6

Число	14	15	16	17	18	19	20	21	22	23
Из чего можно получить число $n$ за один шаг	7, 13	14	8, 16	16	9, 17	18	10, 19	20	11, 21	22
Количество программ, с помощью которых можно получить число $n$ из числа 3 (не обязательно за один шаг)	$2+6=8$	8	$3+8=11$	11	$3+11=14$	14	$4+14=18$	18	$4+18=22$	22

Рис 3.

**Ответ: 22**

Как мы видим из таблицы, количество программ быстро растет с ростом числа  $n$ . Например, для  $n = 24$  получим  $R(n) = 30$ ; для  $n = 26$  получим  $R(n) = 36$ ; для  $n = 30$  получим  $R(n) = 52$ . При заполнении таблицы дополнительная работа невелика, а количество вершин в дереве, которое нужно построить, увеличилось более, чем в два раза.

Задание повышенного уровня сложности В14 существует в КИМ ЕГЭ с 2012 г. Оно предполагает анализ алгоритма, содержащего вызов пользовательской функции. В 2012 и 2013 гг. этими функциями были параболы, в 2014 г. разнообразие функций увеличилось, равно как и формулировки заданий изменились. Задание вызвало явные затруднения экзаменуемых, поэтому стоит привести здесь разбор двух различных по виду заданий этого года.

Первое задание (326 вариант) содержит вызов степенной функции (в приведенном примере это функция  $f=n^3$ ).

**Формулировка задания В14:**

Напишите в ответе число различных значений входной переменной  $k$ , при которых программа выдаёт тот же ответ, что и при входном значении  $k = 64$ . Значение  $k = 64$  также включается в подсчёт различных значений  $k$ . Для Вашего удобства программа приведена на четырёх языках программирования.

Бейсик	Паскаль
<pre> DIM K, I AS LONG INPUT K I = 12 WHILE I &gt; 0 AND F(I) &gt; K     I = I - 1 WEND PRINT I  FUNCTION F(N)     F = N * N * N END FUNCTION </pre>	<pre> var k, i : longint; function f(n: longint) : longint; begin     f := n * n * n end;  begin     readln(k);     i := 12;     while (i&gt;0) and (f(i)&gt;k) do         i := i-1;     writeln(i) end. </pre>

Си	Алгоритмический язык
<pre>#include&lt;stdio.h&gt; long f(long n) {     return n * n * n; }  void main() {     long k, i;     scanf("%ld", &amp;k);     i = 12;     while (i&gt;0 &amp;&amp; f(i)&gt;k) do         i--;     printf("%ld", i); }</pre>	<pre>алг нач     цел i, k     ввод k     i := 12     нц пока i &gt; 0 и f(i) &gt; k         i := i - 1     кц     вывод i кон алг цел f(цел n) нач     знач := n * n * n кон</pre>

**Ответ: 61**

С конкретным заданием справилось всего 18% экзаменуемых, что недостаточно, даже учитывая его новизну.

При решении данной задачи следует выполнить два действия:

Во-первых, определить, что за функция вызывается в основном алгоритме. Для данной задачи это, как хорошо видно из описания, куб числа ( $f=n^3$ ).

Во-вторых, определить, при каком условии завершается выполнение цикла. Для любого целого  $k > 0$  цикл завершается в ситуации, когда  $f(i) \leq k$  (при  $k > 0$  это наступает раньше, чем  $i \leq 0$ ). Следует, таким образом, определить, при каком  $i$  выражение  $i^3 \leq 64$  истинно. Очевидно, что это все  $i \leq 4$ .

Итак, при  $k=64$  программа выведет число 4 (значение переменной  $i$ , такое, что  $i^3 = 64$ ). При  $9 \leq k < 64$  программа выведет число 3, при  $k = 125$  программа выведет число 5. Таким образом, число 4 будет выводиться программой для всех  $64 \leq k < 125$ . Таких целых чисел в этом диапазоне ровно 61.

Другой вариант того же задания (вар. 301) отличается функцией, а формулировки практически совпадают.

#### Формулировка задания В14:

Напишите в ответе число различных значений входной переменной  $k$ , при которых программа выдаёт тот же ответ, что и при входном значении  $k = 120$ . Значение  $k = 120$  также включается в подсчёт различных значений  $k$ . Для Вашего удобства программа приведена на четырёх языках программирования.

Бейсик	Паскаль
<pre>DIM K, I AS LONG INPUT K I = 12 WHILE I &gt; 0 AND F(I) &gt; K     I = I - 1 WEND PRINT I  FUNCTION F(N)     IF N &lt; 2 THEN         F = 1     ELSE         F = N * F(N-1)     END IF END FUNCTION</pre>	<pre>var k, i : longint; function f(n: longint): longint; begin     if n&lt;2 then         f := 1     else         f := n*f(n-1)     end; end;  begin     readln(k);     i := 12;     while (i&gt;0) and (f(i)&gt;k) do</pre>

<pre> END IF END FUNCTION </pre>	<pre> i := i-1; writeln(i) end. </pre>
<b>Си</b>	<b>Алгоритмический язык</b>
<pre> #include&lt;stdio.h&gt; long f(long n) {     if (n&lt;2)         return 1;     else         return n*f(n-1); }  void main() {     long k, i;     scanf("%ld", &amp;k);     i = 12;     while (i&gt;0 &amp;&amp; f(i)&gt;k) do         i--;     printf("%ld", i); } </pre>	<pre> алг нач     цел i, k     ввод k     i := 12     нц пока i &gt; 0 и f(i) &gt; k         i := i-1     кц     вывод i кон алг цел f(цел n) нач     если n &lt; 2         то             знач := 1         иначе             знач := n*f(n-1)     все кон </pre>

**Ответ: 600**

Решение осуществляется аналогично. Проанализировав описание функции, видим, что это факториал ( $f=n!$ ), записанный как рекурсивная функция. Далее мы понимаем, что  $5! = 120$ ,  $6! = 720$  и число 5 программа выведет при всех  $120 \leq k < 720$ . Таких целых значений  $k$  всего 600.

Затруднение вызвало и задание повышенного уровня сложности В8. С ним справились от 14% до 51% экзаменующихся, в зависимости от варианта (средний процент выполнения составляет 26,3%). Отличие этого задания от В14 состоит в том, что оно требует осуществить анализ алгоритма, содержащего вспомогательные алгоритмы, цикл и ветвление, но не содержащего описание функции. В качестве вспомогательных алгоритмов используются хорошо известные экзаменующимся функции целочисленного деления и определения остатка от деления. Алгоритм, описанный в задании, считает количество цифр в десятичной записи числа (значение переменной  $a$  в результате исполнения алгоритма) и произведение этих цифр (значение переменной  $b$ ). Решение задачи предполагает нахождение этих цифр путем поиска делителей числа  $b$ .

**Формулировка задания В8:**

Ниже на четырёх языках программирования записан алгоритм. Получив на вход число  $x$ , этот алгоритм печатает два числа:  $a$  и  $b$ .

Укажите наименьшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 2, а потом 15.

<b>Бейсик</b>	<b>Паскаль</b>
<pre> DIM X, A, B AS INTEGER INPUT X A=0: B=1 WHILE X &gt; 0     A = A+1 </pre>	<pre> var x, a, b: integer; begin     readln(x);     a:=0; b:=1;     while x&gt;0 do </pre>

<pre> B = B*(X MOD 10) X = X \ 10 WEND PRINT A PRINT B </pre>	<pre> begin   a:=a+1;   b:=b*(x mod 10);   x:= x div 10 end; writeln(a); write(b) end. </pre>
<p><b>Си</b></p> <pre> #include&lt;stdio.h&gt; void main() {   int x, a, b;   scanf("%d", &amp;x);   a=0; b=1;   while (x&gt;0){     a=a+1;     b=b*(x%10);     x= x/10;   }   printf("%d\n%d", a, b); } </pre>	<p><b>Алгоритмический язык</b></p> <pre> алг нач   цел x, a, b   ввод x   a:=0; b:=1   нц пока x&gt;0     a:=a+1     b:=b*mod(x,10)     x:=div(x,10)   кц   вывод a, нс, b кон </pre>

В данном случае в исходном числе было две цифры, произведение которых равно 15. Понятно, что эти цифры — 5 и 3. Наименьшим числом, составленным из этих двух цифр, будет число 35.

**Ответ: 35**

И, наконец, 4 задания высокого уровня сложности были призваны выделить выпускников, в наибольшей степени овладевших содержанием учебного предмета, ориентированных на получение высшего профессионального образования в областях, связанных с информатикой и компьютерной техникой, то есть абитуриентов ведущих технических вузов. Выполнение этих заданий давало до 25% от максимального первичного балла. Среди этих заданий одно посвящено основам логики и три – теории алгоритмов и программированию. Одно задание высокого уровня содержится во второй части, а остальные три задания высокого уровня требовали развернутого ответа.

Задание высокого уровня В15, предполагающее краткий ответ в виде натурального числа, является едва ли не самым сложным заданием КИМ ЕГЭ по информатике и ИКТ. С ним, как правило, справляются не более 5% экзаменуемых. Задание проверяет умение преобразовывать выражения, содержащие логические переменные, умение описать на естественном языке множество значений логических переменных, при которых заданный набор логических выражений истинен.

Для того, чтобы выполнить задание, ученик должен уметь:

- преобразовывать логические выражения (включая выполнение замены переменных);
- переводить формальное описание, в виде системы логических условий, на нормальный, "человеческий" язык;
- подсчитать число двоичных наборов, удовлетворяющих заданным условиям.

После того, как выяснено, что за наборы удовлетворяют системе, подсчет их числа относительно прост. Наиболее трудным для усвоения, видимо, является второе из перечисленных требований – оно не формализуется, от ученика, как правило, требуется догадка.

Стоит привести здесь разбор этого задания.

### Пример №1 формулировки задания В15:

Сколько существует различных наборов значений логических переменных  $x_1, x_2, \dots, x_9, x_{10}$ , которые удовлетворяют всем перечисленным ниже условиям?

$$((x_1 \equiv x_2) \vee (x_3 \equiv x_4)) \wedge (\neg(x_1 \equiv x_2) \vee \neg(x_3 \equiv x_4)) = 1$$

$$((x_3 \equiv x_4) \vee (x_5 \equiv x_6)) \wedge (\neg(x_3 \equiv x_4) \vee \neg(x_5 \equiv x_6)) = 1$$

$$((x_5 \equiv x_6) \vee (x_7 \equiv x_8)) \wedge (\neg(x_5 \equiv x_7) \vee \neg(x_7 \equiv x_8)) = 1$$

$$((x_7 \equiv x_8) \vee (x_9 \equiv x_{10})) \wedge (\neg(x_7 \equiv x_8) \vee \neg(x_9 \equiv x_{10})) = 1$$

В ответе **не нужно** перечислять все различные наборы значений  $x_1, x_2, \dots, x_9, x_{10}$ , при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

#### Набросок решения:

Решение состоит из двух этапов. Сначала попытаемся описать, как устроены все наборы значений переменных, удовлетворяющие данной системе. Далее подсчитаем число таких наборов.

*Этап 1. Как устроено множество решений*

А. Предварительный этап – упрощаем уравнения.

В системе фигурируют логические функции от следующих выражений:

$$(x_1 \equiv x_2), (x_3 \equiv x_4), (x_5 \equiv x_6), (x_7 \equiv x_8), (x_9 \equiv x_{10})$$

Подобно тому, как это делается при решении алгебраических уравнений, сделаем замену переменных:

$$t_1 = x_1 \equiv x_2$$

$$t_2 = x_3 \equiv x_4$$

$$t_3 = x_5 \equiv x_6$$

$$t_4 = x_7 \equiv x_8$$

$$t_5 = x_9 \equiv x_{10}$$

Общая формула замены ( $k=1, 2, 3, 4, 5$ ):

$$t_k = (x_{2k-1} \equiv x_{2k})$$

Получим:

$$(t_1 \vee t_2) \wedge (\neg t_1 \vee \neg t_2) = 1$$

$$(t_2 \vee t_3) \wedge (\neg t_2 \vee \neg t_3) = 1$$

$$(t_3 \vee t_4) \wedge (\neg t_3 \vee \neg t_4) = 1$$

$$(t_4 \vee t_5) \wedge (\neg t_4 \vee \neg t_5) = 1$$

Уравнения полученной системы имеют вид ( $k=1, 2, 3, 4$ ):

$$(t_k \vee t_{k+1}) \wedge (\neg t_k \vee \neg t_{k+1}) = 1$$

Это означает, что из каждых двух переменных  $t_k$  и  $t_{k+1}$  ровно одна равна 1 и ровно одна равна нулю, т.е. эти переменные имеют разные значения. Таким образом, систему можно еще немного упростить и записать ее так:

$$\neg(t_1 \equiv t_2) = 1$$

$$\neg(t_2 \equiv t_3) = 1$$

$$\neg(t_3 \equiv t_4) = 1$$

$$\neg(t_4 \equiv t_5) = 1$$

Б. Анализ системы.

В любом решении последней системы значения переменных чередуются. Поэтому такая система имеет ровно два решения: 01010 и 10101 (первая цифра – значение переменной  $t_1$ , вторая – значение  $t_2$  и т.д.).

Далее, т.к.

$$t_k = x_{2k-1} \equiv x_{2k}$$

(здесь  $k=1, 2, 3, 4, 5$ ), то каждому значению  $t_k$  соответствуют две пары значений переменных  $x_{2k-1}$  и  $x_{2k}$ . Например,  $t_k = 1$  в двух случаях:  $\{x_{2k-1} = x_{2k} = 1\}$  и  $\{x_{2k-1} =$

$x_{2k}=0$  }.

*Этап 2. Подсчет числа решений.*

Каждому из двух решений системы для переменных  $t$  соответствует  $2^5 = 32$  решения исходной системы. Поэтому исходная система имеет  $2 \cdot 32 = 64$  решения.

**Ответ: 64**

**Пример №2 формулировки задания В15:**

Сколько существует различных наборов значений логических переменных  $x_1, x_2, \dots, x_{10}$ , которые удовлетворяют всем перечисленным ниже условиям?

$$\neg(x_1 \equiv x_2) \wedge (x_1 \vee x_3) \wedge (\neg x_1 \vee \neg x_3) = 0$$

$$\neg(x_2 \equiv x_3) \wedge (x_2 \vee x_4) \wedge (\neg x_2 \vee \neg x_4) = 0$$

...

$$\neg(x_8 \equiv x_9) \wedge (x_8 \vee x_{10}) \wedge (\neg x_8 \vee \neg x_{10}) = 0$$

В ответе **не нужно** перечислять все различные наборы значений  $x_1, x_2, \dots, x_{10}$ , при которых выполнена данная система равенств. В качестве ответа вам нужно указать количество таких наборов.

**Набросок решения:**

Решение состоит из двух этапов. Сначала попытаемся описать, как устроены все наборы значений переменных, удовлетворяющие данной системе. Далее подсчитаем число таких наборов.

*Этап 1. Как устроено множество решений*

А. Предварительный этап – упрощаем уравнения.

Заметим, что выражение  $(a \vee b) \wedge (\neg a \vee \neg b)$  равносильно тому, что ровно одна из переменных  $a$  и  $b$  равна 1, то есть равносильно выражению  $\neg(a \equiv b)$ . Поэтому каждое выражение вида  $(x_k \vee x_{k+2}) \wedge (\neg x_k \vee \neg x_{k+2})$ , где  $k=1, \dots, 8$ , в наших уравнениях можно заменить выражением  $\neg(x_k \equiv x_{k+2})$ .

Таким образом, наша система эквивалентна системе

$$\neg(x_1 \equiv x_2) \wedge \neg(x_1 \equiv x_3) = 0$$

$$\neg(x_2 \equiv x_3) \wedge \neg(x_2 \equiv x_4) = 0$$

...

$$\neg(x_8 \equiv x_9) \wedge \neg(x_8 \equiv x_{10}) = 0$$

Далее,  $\neg a \wedge \neg b = 0$  означает, что, если  $\neg a$  истинно, то  $\neg b$  истинным быть не может. Т.е.  $\neg a \wedge \neg b = 0$  эквивалентно  $\neg a \rightarrow b = 1$ .

Поэтому систему можно записать в следующем виде

$$\neg(x_1 \equiv x_2) \rightarrow (x_1 \equiv x_3) = 1$$

$$\neg(x_2 \equiv x_3) \rightarrow (x_2 \equiv x_4) = 1$$

...

$$\neg(x_8 \equiv x_9) \rightarrow (x_8 \equiv x_{10}) = 1$$

Б. Анализ системы.

Каждое из уравнений полученной системы имеет вид ( $k = 1, \dots, 8$ ):

$$\neg(x_k \equiv x_{k+1}) \rightarrow (x_k \equiv x_{k+2}) = 1$$

Иными словами, если два соседних элемента набора  $x_k$  и  $x_{k+1}$  не равны между собой, то  $x_k = x_{k+2}$ , то есть элементы  $x_{k+1}$  и  $x_{k+2}$  также не равны между собой. Таким образом, набор удовлетворяет системе, тогда и только тогда, когда он обладает следующими свойствами. В начале набора стоит несколько (может быть, одно) одинаковых значений (назовем это "головой" набора). Затем (после первого появления нового числа) значения в наборе чередуются ("хвост" набора).

Пример решения: 1111010101 (в этой последовательности первая цифра – значение переменной  $x_1$ , вторая цифра – значение переменной  $x_2$ , и т.д.)

Здесь голова набора состоит из четырех единиц, а хвост – это последовательность 01010101. в данном примере длина головы равна 4.

Важное наблюдение. Для каждой непустой головы есть ровно один хвост,



образующий вместе с ней решение. Действительно, первая цифра такого хвоста – это цифра, противоположная цифрам головы. А дальше цифры в хвосте чередуются.

### Этап 2. Подсчет числа решений

В соответствии с важным наблюдением, количество решений совпадает с количеством возможных голов. Очевидно, существует 10 голов, состоящих из единиц (1, 11, 111, ..., 111111111) и столько же голов, состоящих из нулей.

**Ответ: 20**

Как видим, сложность решения задачи не зависит от числа переменных и уравнений. Если понятно, как устроено множество решений, подсчитать количество решений для аналогичной системы, скажем, с 20-ю переменными, не сложнее, чем в уже рассмотренном случае.

Отметим, что разбирать эту задачу стоит только с учениками, которые достаточно свободно владеют преобразованиями логических выражений. Стоит выписать несколько полезных преобразований (они встречались в разобранных примерах):

$$\begin{array}{ll} \neg a \vee b & \text{равносильно } a \rightarrow b \\ (a \rightarrow b) \wedge (a \rightarrow b) & \text{равносильно } a \equiv b \\ (\neg a \vee b) \wedge (a \vee \neg b) & \text{равносильно } a \equiv b \\ (a \vee b) \wedge (\neg a \vee \neg b) & \text{равносильно } \neg(a \equiv b). \end{array}$$

Кроме того, полезно потренироваться в выполнении замен в логических выражениях. Отметим, что это делается точно так же, как и замены в уравнениях, которые встречаются в курсе математики.

Первая цель при выполнении задания В15 - понять, что собой представляет множество решений системы. Для этого систему бывает полезно преобразовать (упростить) систему, используя тождественные преобразования и замены переменных. Затем – подсчитать количество элементов во множестве решений. Во многих случаях система состоит из однотипных уравнений, каждое из которых связывает небольшое число переменных (две-три-четыре), при том, что в системе может быть 10 и более переменных. Обычно, количество переменных не является источником сложности, оно является параметром решения. Если не получается решить задачу в общем виде, можно попробовать перебрать все решения для системы с небольшим количеством переменных. Это может подсказать, как выглядит решение в общем виде.

Если понятно, как выглядит множество решений, подсчет их количества – несложная комбинаторная задача. Сильные ученики могут сообразить, как провести подсчет, даже не обладая специальными знаниями. Стоит повторить формулы произведения возможностей и формулу суммы арифметической прогрессии.

### **Можно дать следующие общие рекомендации по подготовке к ЕГЭ 2015 г.**

Во-первых, ЕГЭ по информатике – экзамен по выбору, но, тем не менее, выпускников, сдающих этот экзамен, по уровню исходной подготовки можно разделить на три группы: базовый уровень; хороший уровень и отличный уровень.

Экзаменуемые с базовым уровнем подготовки, как правило, выполняют только задания базового уровня сложности (от 10 до 15 таких заданий). Эти задания покрывают (на базовом уровне) основной материал курса информатики и ИКТ, в том числе, темы «Двоичное представление чисел», «Файловая система персональных компьютеров», «Базы данных», «Электронные таблицы», «Кодирование текстовой информации», «Основы логики», «Основы теории алгоритмов». Работа экзаменуемых этой группы происходит, в основном, на уровне воспроизведения и применения знаний в стандартной ситуации, Абитуриенты этого уровня будут испытывать затруднения при изучении информатики и ИКТ в профильных учреждениях высшего профессионального образования, связанные с недостаточной подготовкой.

При подготовке к ЕГЭ ученикам этой группы разумно сосредоточиться на темах, проверяемых упомянутыми выше заданиями (А1-А9, В1-В6). Более амбициозная (и более предпочтительная) стратегия состоит в том, чтобы более глубоко изучить курс в целом и поднять свой уровень знаний с базового до хорошего.

Экзаменуемые с хорошим уровнем подготовки, как правило, выполняют все задания, кроме наиболее сложных заданий (В14, В15, С2, С3, С4). Говоря о группе в целом, следует обратить внимание на тему «Алгоритмы и программирование», которой посвящены 4 из 5 наиболее трудных задач ЕГЭ 2014 г. Кроме того, ученики этой группы имеют, как правило, пробелы в отдельных темах – индивидуальные для каждого ученика. Эти проблемные темы должны быть своевременно выявлены с помощью тестовых работ и тщательно разобраны. Подчеркнем, что речь идет именно об изучении тем, а не о натаскивании на отдельные задания.

Экзаменуемые группы с отличным уровнем подготовки показывают хорошее знание всех разделов курса информатики и ИКТ и готовность к продолжению образования на профильных специальностях учреждений высшего профессионального образования. Однако, даже среди этой группы процент выполнения задачи С4, в которой требуется самостоятельно написать программу, достаточно низок (некоторым оправданием этого является то, что разработка программы ведется на бумаге, без использования привычной программной среды). Таким образом, резерв в повышении результатов в этой группе состоит в изучении программирования (задача С4) и уменьшении количества потерянных баллов в других задачах.

Во-вторых, следует учитывать неравномерность изучения различных тем в образовательных учреждениях. Рассматривая отдельные темы, можно отметить хорошее, в целом, решение экзаменуемыми заданий по темам «Системы счисления» и «Файловая система». Следует обратить внимание на изучение понятийного аппарата и основных формул, связанных с организацией и функционированием компьютерных сетей, передачей данных, кодированием звуковых и графических данных. При обучении на профильном уровне следует больше внимания уделять разработке программ в рамках требований, предъявляемых стандартом образования (задача С4). Как при профильном, так и при базовом обучении, следует максимальное внимание уделять решению задач, в том числе – решению практических задач на построение алгоритмов с помощью компьютера. Большую помощь при этом могут оказать практикумы, включающие наборы задач по разным темам и допускающие выполнение обучающимися самопроверки.

В целом, при обучении важно обращать внимание учащихся не только на простые факты и формулы, но и на более глубокие связи между объектами и понятиями. Для проверки глубины освоения материала в ЕГЭ есть задания (А1, А3, А8, В5, В7, В9, В13, В14, В15), решение которых, основанное на понимании особенностей задачи, существенно проще решения, основанного на прямом применении заученных формул.

При подготовке выпускников к Единому государственному экзамену учителям следует подробнее объяснять учащимся цели этого испытания и структуру экзаменационной работы. Так как экзамен используется и для оценки уровня усвоения образовательной программы и соответствия подготовки выпускников требованиям государственного стандарта образования с одной стороны и для ранжирования подготовки абитуриентов к продолжению обучения на профильных специальностях вузов с другой стороны, экзаменационная работа содержит набор заданий различной сложности и преследующих различные цели. Будущему участнику экзамена надо четко определиться с тем, какие цели он ставит и, соответственно, в какую из групп по уровню результатов планирует попасть. Как показывают результаты экзамена, только небольшая часть его участников показали

необходимый для продолжения образования на профильных специальностях уровень подготовки. Эта подготовка включает умение использовать электронные таблицы для обработки статистических данных, в том числе результатов научных исследований, умение самостоятельно разрабатывать программы на языках программирования для решения практических задач обработки массивов данных, умение использовать ресурсы Интернет для поиска и систематизации информации. Поэтому следует обратить внимание на такие разделы кодификатора содержания как 1.3.2 (Математические модели), 1.5.2 (Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы), псевдослучайные последовательности), 1.5.6 (Сортировка), 3.4.1 (Математическая обработка статистических данных), 3.5.2 (Использование инструментов поисковых систем, формирование запросов). Надо иметь в виду, что учреждения высшего профессионального образования заинтересованы в абитуриентах, чья подготовка соответствует следующим требованиям кодификатора требований: 1.1.1 (Проводить вычисления в электронных таблицах), 1.1.5 (Создавать программы на языке программирования), 2.9 (Проводить статистическую обработку данных с помощью компьютера).

Следует понимать, что требования учреждений высшего профессионального образования к подготовке абитуриентов профильных специальностей предполагают уровень подготовки, соответствующий профильному курсу информатики и ИКТ, поэтому учащимся с базовой подготовкой не следует рассчитывать на высокий результат ЕГЭ. Задания базового уровня обеспечивали экзаменуемому максимум 15 первичных баллов, что соответствует «удовлетворительному» уровню, недостаточному для продолжения образования на профильной специальности.

При подготовке к ЕГЭ задания демонстрационной версии экзамена надо рассматривать только как ориентиры, показывающие примерные образцы заданий, которые могут стоять на соответствующих позициях. Натаскивание абитуриентов на определенный тип заданий может привести к тому, что они будут обескуражены заданием, немного отличающимся по формату от того, которое было ими многократно разобрано.

Методическую помощь учителям и обучающимся при подготовке к ЕГЭ могут оказать материалы с сайта ФИПИ: [www.fipi.ru](http://www.fipi.ru):

– документы, определяющие структуру и содержание КИМ ЕГЭ 2015 г. (кодификатор элементов содержания и требований к уровню подготовки выпускников, спецификация и демонстрационный вариант КИМ);

– Открытый банк заданий ЕГЭ;

– учебно-методические материалы для председателей и членов региональных предметных комиссий по проверке выполнения заданий с развернутым ответом экзаменационных работ ЕГЭ;

– аналитические отчеты о результатах экзамена, методические рекомендации и методические письма прошлых лет.