

ЭКЗАМЕН ПО ИНФОРМАТИКЕ. Как к нему подготовить?

Ирина Анатольевна Милютина,
*ведущий инженер-программист Лаборатории медиаобразования
Института содержания и методов обучения РАО,
учитель информатики гимназии № 625 г. Москвы*

Экзамен по информатике перешёл в формат ЕГЭ. У российских школьников появилась возможность аттестации и по этому учебному предмету. Однако на этом пути возникают трудности. Как наиболее эффективно организовать такую подготовку?

• информатика • требования вузов • проблемы сдачи ЕГЭ • приёмы программирования • учебники • образовательные стандарты

Единые правила «игры»

Наконец мы дождались экзамена по информатике в форме ЕГЭ. Сразу хочу сказать, что я сторонник этой формы по целому ряду причин, основные из которых — чёткие и понятные правила «игры», то есть правила проведения экзамена, перечень тем, уровни сложности заданий и др.; равные условия для всех сдающих экзамен, независимо от прохождения и оплаты курсов вуза, статуса родителей и т.п. Не знаю, многие ли пробовали подготовить учеников к экзамену по информатике в условиях, когда каждый вуз, который его принимал, выдвигал свои собственные требования. Подготовить всех в соответствии с требованиями даже основных вузов было большой проблемой, да и смысла в этом не было в рамках традиционных уроков информатики. Сейчас, с введением ЕГЭ, правила одни для всех, что очень важно.

Второй положительный момент, который хочется отметить: введе-

ние независимой экспертизы дисциплинирует учащихся и учителей, стимулирует полное прохождение тем курса. Раньше учитель мог часть тем вообще не проходить или пройти на элементарном уровне, недостаточном для решения даже простейших задач (это косвенно подтверждает и самый высокий процент неудовлетворительных оценок на экзамене среди прочих предметов, сдаваемых в форме ЕГЭ в 2009 году). Немало учителей сами не владеют темами курса, особенно в областях программирования и алгоритмизации, использования электронных таблиц или алгебры логики. И эту ситуацию никто вообще-то не контролировал.

Все проблемы, связанные с неполным прохождением школьного курса информатики, ложились на плечи учащихся в вузах, где они вынуждены были компенсировать отставание. Особенно эта нехватка знаний проявлялась в технических и экономических специальностях, где вчерашний школьник должен был не столько осваивать новые технологические приёмы решения задач в рамках определённого профессионального программного обеспечения, сколько сначала научиться алгоритмически

и логически мыслить, подбирать технологии из множества, знать и понимать возможности технологий, уметь их грамотно реализовывать, оптимизировать решения задач за счёт правильно подобранных технологий. Ликвидировать такое отставание очень сложно, а иногда и невозможно. Отметим, что экзамен по информатике ввели именно технические вузы (это около 50% специальностей), что показывает высокую степень их заинтересованности уже на момент поступления абитуриентов, владеющих основами пошаговой детализации задач, алгоритмизации и программирования, понимающих и умеющих выполнять логические операции, анализировать их.

Структура единого экзамена достаточно гибка для наращивания сложности заданий, для корректировки имеющихся и для включения в перечень заданий новых тем, в зависимости от потребностей общества. Это тоже безусловный плюс.

Как сдать экзамен?

1. Отсутствуют ясно и чётко сформулированные цели курса информатики, что приводит к противоречию между содержанием программы по информатике и содержанием заданий ЕГЭ. Собственно, что мы желаем получить на выходе учащихся из стен школы? Людей, умеющих грамотно и быстро получать, обрабатывать, всесторонне анализировать информацию и сохранять её в исходной или переработанной форме? Тогда необходимо выбрать основные, требующиеся широкому кругу пользователей технологии, включая технологии анализа информации по содержанию, чего нет и не было в курсе информатики, отрабатывать рациональные приёмы работы с информацией, экономящие время, приводящие к правильным решениям при минимуме затрат. Это чрезвычайно важные задачи современного общества. Или мы желаем выпускать учащихся специалистами, умеющими самостоятельно разработать технологии решения задач? Это направление потребует знаний технологии программирования, алгоритмизации, структурирования задач, развития логики на высоком уровне. Задания единого экзамена явно поддерживают второе направление. А нужно ли это всем, и если да, то как глубоко? Из-за этой неразберихи в целях и задачах курса информатики появились перечисленные ниже проблемы, связанные с программированием, при сдаче экзамена.

2. Тему моделирования необходимо жёстко увязать с программированием, так как в основе программы лежит модель, а учащиеся не умеют самостоятельно строить модели, формализовывать полученные текстовые задания, структурировать процесс решения и т.п. Эта же проблема очевидна и при изучении электронных таблиц, где также важнее научиться не просто считать по готовым и известным формулам, а самим создавать эти формулы, что совершенно игнорируется в стандарте и в учебниках по информатике.

3. Обучать программированию (а три из четырёх заданий группы С связаны именно с программированием) для полноценной сдачи экзамена достаточно сложно, нужно уделять этой части предмета значительно больше времени, чем предусматривает программа. Необходимо изучать не только основные конструкции языка программирования, но типовые алгоритмы решений, подходы к рациональному использованию памяти и минимизации времени работы программ, организацию проверки неверно введённых или полученных данных, типы данных и области их применения и многое другое. Без этого обучение программированию теряет всякий смысл. Дети думают, что научились программировать, а простейшие практические задачи их ставят в тупик.

4. Основа любого обучения программированию — большое количество практических задач, которые должен выполнить учащийся самостоятельно. Одно или два практических занятия по каждой из алгоритмических конструкций — это всё равно, что ничего. При таком объёме практики (а базисный учебный план другого объёма и не предусматривает без ущерба другим темам курса) говорить об изучении основ программирования просто смешно.

5. Способность находить рациональные алгоритмы решения и создавать качественные программы является профессиональной, её имеют совсем не все учащиеся

(и это нормально), а стандарт образования и программа не подразумевают уровней освоения, автоматически задавая всем учащимся единые требования. Конечно, можно уповать на профильное обучение, но при отсутствии перечисленного в предыдущих пунктах это тоже не даёт нужного результата.

6. Учащиеся должны чётко представлять, какие из специальностей потребуют знания основ программирования. Учащимся, ориентирующимся на гуманитарное образование, в своей дальнейшей жизни вполне достаточно будет уметь выстраивать технологический процесс решения задачи, выделять подзадачи или стандартные блоки её решения, подбирать технологии из имеющихся для реализации каждого блока. Однако на уровне средней ступени школы рано судить о будущей профессии школьника, поэтому основы алгоритмизации и программирования преподавать необходимо всем. Тогда возникает вопрос: если учить всех, то будущие гуманитарии будут находиться в сильном затруднении и ощущать себя неуспешными учениками, что может не соответствовать действительности, а будущие «технари» не получат возможности сделать более разнообразным круг решаемых задач, которые им необходимы в дальнейшем.

Проблемы решаются

Как выйти из такого положения? Подчеркну, что при отработке приёмов программирования речь не может идти о простом наращивании сложности задач для успевающих учеников. Отрабатывается большое количество стандартных приёмов, каждый из которых необходимо объяснять, а каждому ученику самостоятельно их опробовать. Какой же напрашивается вывод? Сейчас мы подходим к главному — новой версии образовательного стандарта по информатике и ИКТ. Нужно не просто добавить новые направления в стандарт образования или скорректировать уже имеющиеся, необходимо принципиально изменить подход к формированию данного стандарта. Надо чётко сформулировать минимум знаний и умений, который будет не-

обходим каждому ученику, независимо от его будущей профессии и специальности. Сейчас этот минимум более или менее сформирован для информационно-компьютерных технологий, причём в основном это касается только работы с текстом.

Для каждой крупной темы (раздела) стандарта за обязательным минимумом должны следовать несколько уровней достижений для школьника, к которым он будет стремиться, которые будут показывать степень освоения им той или иной технологии, его потенциальный профессионализм в каждой технологической области. Переход на каждый новый уровень должен поощряться и приветствоваться. Но если ребёнок чувствует, а учитель констатирует объективные затруднения с освоением нового более высокого уровня какой-либо технологии, можно остановиться на обязательном минимуме и достигать совершенства в его использовании, а также расширять круг осваиваемых технологий, более близких и понятных данному конкретному учащемуся, например, в рамках проектных заданий. Следствием усвоения обязательного минимума не должны быть снижение оценок или другие карательные санкции, просто каждый уровень будет ориентировать школьника в степени профессиональной готовности пользоваться технологией, в степени готовности к сдаче единого экзамена по информатике в полном объёме и, в конечном итоге, давать объективную картину для выбора будущей профессии в информационном пространстве. Ни в коем случае стандарт не должен содержать формулировок типа «представление о языках программирования» или «знакомство с одним из языков программирования», так как представлять и знакомиться можно тоже по-разному. Требования должны быть чёткие и понятные.

Мне представляется серьёзной ошибкой попытка наращивать темы стандарта всемирными новейшими технологиями, которые появились в настоящее время. При этом подходе ни одна из технологий не получает должного освещения и представление о ней и её возможностях или минимальное или

вообще превратное. Особенно страдают от такого подхода темы, требующие алгоритмического мышления и знаний в области работы с памятью компьютера (электронные таблицы, базы данных и т.п.). В вузах основную часть вчерашних школьников приходится отучать от неправильно заложенных технологических приёмов и решений, которые совершенно не были заметны на условных примерах школьного курса, но приводят к огромным потерям времени и сил при реальных практических задачах. Гораздо правильнее объяснить ученикам общие подходы, которые используются разработчиками технологий для формирования пользовательской среды оболочек, правила получения справочной информации и т.п. Такой подход позволит ученикам легче осваивать появляющиеся новые технологии, быстрее и эффективнее изучать их возможности, а попытка включать в курс информатики изучение новых технологий практически «с нуля», безгранично расширять их состав, приведёт к классическому результату — ученики будут «ничего не знать обо всём». Необходимо изучать в обязательном порядке только технологии, массово используемые на практике, а это, безусловно, технологии, входящие в офисные пакеты, плюс элементы графики. Насыщение школьного курса технологиями в области развлечений также приводит к желанию школьников развлекаться, «прикалываться», но не работать осмысленно. Любое затруднение вызывает бурное неприятие технологии и желание перейти к знакомым, привычным и не обременяющим мозг развлечениям.

Честно говоря, серьёзные претензии у меня не только к образовательным стандартам, но также к программам по информатике в школах и, естественно, к учебникам. Сейчас мы имеем достаточно серьёзные противоречия между тем, чему можно научить школьника с помощью всего вышперечисленного, и тем, что реально понадобится не только для сдачи экзамена, но и в нашей обыденной непростой жизни. Из компьютерных технологий, которые должны освоить учащиеся в школе на приемлемом для работы с ними уровне, более или менее они усваивают приёмы работы с текстом и с текстовым редактором (процессором). Электронные таблицы, базы данных, презентационные программы часто представлены в школьных программах и учебниках на уровне, существенно ниже минимально необходимого, даже для того чтобы разобраться,

зачем, собственно, эта технология придумана и где она используется. Кроме того, повторяюсь, привычка работать с условными, а не реальными задачами и данными приводит к тому, что в школьных учебниках появляется масса нерациональных или неправильных технологических приёмов, которые закрепляются в памяти вчерашних школьников и потом очень тяжело оттуда искореняются.

Например, увлечённость спецэффектами при создании и оформлении презентаций, которые не только затрудняют, но часто делают невозможным восприятие информации, заложенной в презентации. Под видом презентаций чаще всего делают клипы. При этом практически нигде нет данных о том, какие типы презентаций существуют, каковы их особенности; сколько и какого вида объектов может находиться на одном слайде; как отобрать информацию для презентации; как выполнить управляемую и настраиваемую на различную аудиторию презентацию; как проводить репетицию презентации и что на ней необходимо выяснить; психологические особенности восприятия информации и их учёт при разработке презентации; продолжительность презентаций разных видов и многое другое. Последствия непродуманного подхода к созданию презентации — вращающиеся рисунки и абзацы текстов, полное заполнение слайдов мелким текстом, различные «вылеты», «растворения», «мельницы» и т.п. (что очень утомляет зрителей, особенно, если информация серьёзная) для каждого из элементов слайда, нагромождение картинок, которые не видны не только в актовом зале, но и с первого ряда парт в классе. Нет самого главного — чёткого осмысления цели создания каждой презентации, «привязки» технологических приёмов к соответствующему типу презентации. В электронных таблицах, в частности Excel, совершенно не изучаются правила формульного и функционального моделирования, если задача предполагает самостоятельное конструирование формулы и сочетание нескольких

функций — это для школьника огромная проблема, часто неразрешимая. Типовая грубая ошибка — выравнивание числовых данных с помощью кнопок выравнивания, в то время как его следует производить относительно десятичной точки (или запятой). Другая типовая ошибка — набор на клавиатуре имён ячеек функций и др. Все эти ошибки могут привести к большим проблемам и потере времени при переходе на значительные объёмы данных. Совершенно не уделяется внимания функциям, кроме самых простых, вложениями функций друг в друга, заполнению диапазонов данными, кроме распространения. Естественно, нет ни слова про обработку ошибочных ситуаций (это также очень важно и в программировании), работу с массивами данных, поименованными переменными и областями их действия. Электронные таблицы вообще следует проходить после темы «Основные алгоритмические конструкции» и желательно вместе с основами программирования или близко по времени, так как решение задач будет иметь аналогичные подходы.

Базы данных чаще всего подразумевают изучение основы работы в Access, в то время как в рамках школы практически нет реальных примеров для грамотной иллюстрации её возможностей. Сразу хочу сказать, что условные примеры следует, по возможности, отбросить, они только способны запутать школьника. Каждая технология должна быть в школе уместна. Базы данных вполне можно конструировать в рамках Excel, тем более что около 90% баз данных табличные. Это не потребует изучения ещё одной технологии, но и позволит достаточно полно дать материал о создании базы, работе с запросами, подготовке выборки по разному набору критериев и т.п. Сейчас практически нет в учебниках внятного объяснения, какие данные должны включаться в запись создаваемой базы, а какие целесообразно вынести в другую, по каким критериям это делается; в реальных задачах именно это очень важно. Кроме того, базы данных также можно объединить во времени с прохождением в области программирования темы «структурированные типы данных, записи». Это даст возможность лучше понять данную тему в программировании, а также определит область, где используются записи. Особое не-

приятие вызывает подача в учебниках темы «Программирование». Во-первых, нельзя изучать несколько языков программирования сразу, это только путает всех, включая учителя; во-вторых, не следует сразу изучать объектный язык без соответствующего ему структурного (кроме того, структурные языки понятнее школьнику, а это, как я уже отмечала, и так одна из самых сложных тем); в-третьих, нельзя иллюстрировать новый материал (даже основные алгоритмические конструкции) сразу с помощью достаточно сложных задач, нужно отрабатывать простые и типовые; в-четвёртых, необходимо сразу обучать школьников организации проверки правильности ввода исходных данных, обработке ошибочных ситуаций, иначе у них складывается впечатление, что об этом позаботится сам компьютер или кто-то другой. Наиболее приемлемое решение данной проблемы — выпуск и использование в школе отдельных от информатики учебников по программированию (можно по алгоритмизации и программированию), кроме того, можно разрешить школам пользоваться старыми, хорошо и многократно проверенными учебниками и учебными пособиями (выпуска ранее 2000 г.).

В современном мире умение работать с информацией — залог успеха, информационное поле разрастается с огромной скоростью, порождает новые проблемы и ставит новые задачи. Тратить в этих условиях драгоценное время вчерашних школьников на переучивание, освоение второстепенного без осмысления основного — чудовищное расточительство.

Итак, стандарт образования и программы, а также учебники выстроены так, что школьник до последнего звонка может не знать, что предмет «информатика» — это про одно, а экзамен — про другое. Иллюзии, что, умея печатать и грамотно находить информацию в сетях, имея смутное представление об электронных таблицах, базах данных, алгебре логики и программировании, можно сдать экзамен и без проблем учиться в техническом или экономическом вузе, обходятся школьникам, вузам и, в конечном итоге, обществу очень дорого. **НО**