

## **СРЕДСТВА ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ КАК СВЯЗУЮЩИЙ ЭЛЕМЕНТ STEM-ОБРАЗОВАНИЯ**

Каган Эдуард Михайлович (sinner@idp-crew.com)

Государственное автономное образовательное учреждение высшего образования города Москвы «Московский городской педагогический университет» (ГАОУ ВО МГПУ)

### **Аннотация**

В статье рассматриваются ряд проблем, связанных с текущим уровнем изучения основ программирования в рамках школьного курса, с позиции STEM-образования и возможности применения современных визуальных сред программирования как интегрирующего компонента для организации междисциплинарного взаимодействия.

Согласно концепции долгосрочного социально-экономического развития РФ до 2020 года основной целью является доведение социального и экономического уровней развития до показателей первых стран мира. Такое стратегическое решение, очевидно, требует не только экономических и социальных реформ, но и во многом затрагивает сектор образования, так как формирование нового социального уклада необходимо начинать именно с образования, а технические новшества (изобретения, новые технологии и открытия) также являются следствием образования.

Таким образом, одной из наибольших проблем, на данный момент препятствующих устойчивому развитию экономики, является отставание развития образовательной среды. На протяжении уже более чем десяти лет организации, занимающиеся формированием учебных программ по всему миру, работают над решением проблемы повышения качества инженерного образования и повышения уровня технологической культуры. Одним из результатов этой работы стала новая методика обучения STEM [1], основанная на четырех ключевых академических дисциплинах: науке, технологии, инженерии и математике — и постулирующая повсеместное применение междисциплинарного обучения.

Несмотря на то что такого рода инициатива является ответом на запрос рынка, ее применение также может привести к решению многих проблем современной системы образования. При использовании междисциплинарного подхода знания учащихся будут более практическими, а сам процесс обучения более интенсивным, так как при изучении материала будет необходимо постоянно возвращаться и использовать навыки иных дисциплин. Одновременное изучение материала из разных областей позволит учащимся быть более уверенными в собственных знаниях, так как количество ассоциаций также будет увеличиваться.

Наличие острого ума, уверенность в собственных силах и знаниях — это путь к профессиональному и техническому творчеству, что необходимо для успешности современного человека в обществе, так как уровень личной технологической вовлеченности будет расти. Таким образом,

наличие междисциплинарных занятий, поддерживаемых технологиями, необходимо для формирования инженерного мышления у учащихся. С этой точки зрения STEM является крайне перспективным направлением, так как позволяет сформировать единую образовательную среду, направленную на формирование универсальных образовательных действий (УУД).

Использование междисциплинарного подхода помогает воспитать у учащихся критическое мышление, которое в дальнейшем позволит обрабатывать разнородную информацию, проводить анализ, приходиться к выводам и ставить новые цели. На данный момент наиболее перспективным основанием для введения элементов STEM-образования является «Информатика и ИКТ» [4], так как именно в рамках данной дисциплины происходит наибольшее размытие границ между областями знаний. К сожалению, отсутствие постоянного вовлечения информационных технологий и работы по формированию системного стиля мышления в рамках многих предметов снижает эффективность обучения.

Также необходимо отметить, что в последние годы наметилась тенденция смещения акцента с изучения информатики и основ программирования на изучение информационных технологий. При этом упускается, что именно работа над алгоритмами крайне важна для формирования фундаментальных навыков системного мышления, таких как декомпозиция задачи, структурирование работы, самостоятельный поиск информации, а также формирование алгоритма решения ранее неизвестной задачи (то есть автономного формирования и закрепления навыка).

Учитывая такое смещение и необходимость поддержки курсов других дисциплин, следует признавать необходимость новых методик и инструментов, используемых в образовании, так как текущая модель изучения программирования становится все менее применимой. Важно заметить, что материал должен не только удовлетворять требованиям и формировать УУД, но и быть легким в усвоении всеми категориями обучающихся и интегрируемым с другими курсами. Классический подход к изучению индустриальных языков программирования, таких как Pascal, Basic, Java, Delphi или C/C++, в данном контексте имеет ряд недостатков.

Первый и наиболее существенный с точки зрения STEM недостаток: курс не во всех случаях формирует понимание, что программирование есть такой же инструмент, как линейка или циркуль, и может быть применен где угодно. Вместо этого программирование зачастую воспринимается как отдельный набор, не имеющий ничего общего с повседневной жизнью. Как следствие, ученик даже не задумывается о применении навыков в других дисциплинах.

Вторым недостатком является текстовая природа языков, так как обучающемуся приходится работать не с наглядным материалом. Формирование даже простой симуляции для проведения опыта по физике на одном из вышеперечисленных языков является задачей для профессионального программиста, а не для школьника. Для опытного программиста привычно выражать мысли в рамках синтаксиса и с учетом

семантики языка программирования, а для школьника это лишь дополнительная трудность, мешающая решению задачи, что снижает как мотивацию, так и продуктивность.

Третий недостаток касается фундаментального вопроса — ориентации курса. Смещение акцента на информационные технологии говорит о том, что курс ориентирован на среднестатистического ученика, который не занимается программированием всерьез. Однако, согласно текущим реалиям, необходимо каждому ученику привить основы системного мышления, чего отдельно изучаемые программы сделать не могут, так как в каждой из них будет своя логика и правила, и достаточно будет заучить последовательность действий, чтобы получить хороший балл. Систематизировать изученные технологии и применять несколько инструментов совместно, постепенно продвигаясь к решению, для школьника является зачастую непосильной задачей. По этой причине знания остаются фрагментированными, и не происходит их интеграция в повседневную жизнь.

Итак, существует ряд трудностей, которые могут быть решены посредством STEM. Как уже было указано выше, STEM изначально задумывался как подход, при котором будет повышаться количество учеников, готовых и стремящихся продолжить обучение в области инженерных наук.

Для этого необходимо развитие навыков использования знаний из разных областей для решения задачи, при этом связующим звеном может служить именно изучение основ программирования, так как оно позволяет работать с абстрактными понятиями, легко соединять знания из разных областей и быстро получать наглядный результат в виде компьютерной симуляции. Важно, что программирование в данном случае понимается не как процесс формирования программы для ЭВМ, а больше как процесс формирования алгоритма и проведения наглядной симуляции.

Яркими примерами вспомогательных средств могут служить специализированные учебные среды и языки программирования Scratch! и Google Blockly. Программирование в них сводится к манипуляциям с графическими объектами, что значительно нагляднее исходного кода. Избавление от исходного кода также ускоряет создание программы, так как защищает от синтаксических ошибок. Формирование программы по принципу сборки конструктора позволяет ученику сконцентрироваться на структуре алгоритма, а не на его записи.

Также, кроме непосредственного программирования, ученик имеет возможность работать с графическими объектами, что позволяет создавать наглядные и сложные симуляции. Ярким примером может быть использование Scratch-программы для демонстрации закона Ома при проведении урока по физике [2]. Такого рода сопровождающий материал позволит ученикам не только лучше понять физический процесс, но и наглядно продемонстрирует применение программирования как инструмента для проведения вычислений, симуляции процессов или же подготовки демонстраций.

Важно заметить, что такого рода среды не только полезны для проведения междисциплинарных занятий, но и позволяют повысить качество освоения базового курса информатики. Так, например, набор заданий Blockly Games позволяет освоить основные приемы программирования, такие как последовательность, ветвление и цикл [3]. Каждая задача требует от ученика создать или скорректировать небольшую графическую программу. Обучение происходит в игровой интерактивной форме, что также повышает вовлеченность в процесс. При этом не требуется установка дополнительного программного обеспечения, задания можно выполнять и во внеурочное время посредством сети Интернет.

Таким образом, появление новых средств и языков программирования, основанных на манипуляции с графическими объектами, позволяет в значительной степени увеличить интегрирующую составляющую изучения основ программирования, а также применять само программирование как интегрирующую активность. Blockly, Scratch и подобные среды могут содействовать развитию системного мышления за счет того, что позволяют ученику не отвлекаться на особенности записи программы, а работать непосредственно над формированием алгоритмической и визуальной составляющими решения.

#### Литература

1. Чемяков В.Н., Крылов Д.А. Stem – новый подход к инженерному образованию // Вестник Марийского государственного университета. – 2015. – № 5. – С. 59-64.
2. Marji, Majed «Learn to Program with Scratch». – No Starch Press, 2014. – 288 p.
3. Семионенков М.Н. Графическая среда программирования Blockly (Блокли) / М.Н. Симеоненков // Информатика. – 2014. – № 3. – С. 32–40.
4. Копосов Д.Г. STEM образование на уроках информатики. [Электронный ресурс]. – Режим доступа: [http://koposov.info/?page\\_id=2097](http://koposov.info/?page_id=2097)