

Д. В. Андреев, Л. М. Изосимова

Аппаратное и программное обеспечение курса робототехники на основе технологий LEGO

В статье проводится обзор существующего аппаратного и программного обеспечения, необходимого для организации курса робототехники на основе технологий LEGO, в задачи которого входит обучение основам программирования и информационных технологий. Далее рассматривается пример курса робототехники, описываются его основные разделы, предлагаются рекомендации по организации работы с программным обеспечением. Также проанализированы наиболее распространенные трудности, связанные с интеграцией курса программирования в курс робототехники.

Ключевые слова: LEGO Mindstorms, программирование, робототехника.

D. V. Andreev, L. M. Izosimova

Organisation of the Course of Programming as an Integral Part of the Course of Robotics Based on LEGO Technologies

This article describes the existing hardware and software, necessary for the organization of the course of robotics based on Lego technologies, which contains the task of teaching the basis of programming and information technologies. Then it considers the example of the course, describes the main sections of the course and offers recommendations for the organization of the work with the software. The most common difficulties with integration of the course of programming with the robotics are also described.

Keywords: LEGO Mindstorms, programming, robotics.

Лего-конструкторы – развивающая среда для детей разного возраста и возможность реализации задач различного плана. Для самых маленьких – это развитие мелкой моторики, работа с цветом, формой, тренировка памяти и внимания. Далее – знакомство с устройством отдельных узлов и механизмов. И, наконец, реализация проектной деятельности, конструирование и программирование самостоятельно разработанной модели, ее тестирование, апробация, представление и презентация.

Конструкторы Лего создают среду, способную обеспечить ребенку живой, естественный, всесторонний образовательный и воспитательный процесс. Подобно изучению родного языка в условиях непосредственного непринужденного и каждодневного общения, познание окружающего мира с Лего становится тем естественным процессом активного практического творчества, когда новый опыт, новые знания ребенок получает легко и радостно в процессе игры, стремясь усовершенствовать мир доступными и понятными ему средствами. «Любой зодчий нуждается в материалах, из которых он будет строить». Лего вполне может быть той культурной средой, в которой многие дети успешно и с энтузиазмом могут «возводить структуры собственного интеллекта». Лего-творчество, затрагивая эмоциональную сферу ребенка, придает его познавательной деятельности личностный характер и тем самым делает ее более плодотворной.

Эти идеи философии «конструкционизма» С. Пайперта в их противопоставлении «инструкционизму», связанному с «формально организованными сферами обучения», на наш взгляд, соответствуют духу времени.

Стихийное знакомство ребенка с Лего или индивидуальные спланированные занятия с ребенком могут начаться с совсем раннего возраста, в то время как системные занятия в группе, имеющие целью постепенное и неуклонное включение учебного компонента в игровую деятельность детей, требуют сформированности навыков группового взаимодействия, концентрации внимания. Поэтому мы считаем оптимальным привлекать к занятиям по Лего-конструированию и программированию учащихся 3–4 классов, уже имеющих опыт работы в коллективе, проявляющих стойкий интерес к Лего, склонных к усложнению игровой деятельности. В связи с этим формирование группы для успешных

занятий предполагает предварительную беседу с будущими учениками об их опыте Лего-конструирования и нежесткий отбор.

К 9–10 годам некоторые дети уже имеют навыки Лего-конструирования, и в этом случае перейти к более серьезным, системным занятиям можно легко и естественно. Важно встретиться с детьми, воспринимаящими Лего как «культурное окружение», в котором им интересно играть, учиться, фантазировать, ставить задачи, искать и находить возможности их решения.

Хорошо, когда управляемые роботы Лего органично вписываются в привычный круг уже знакомых и понятных детям исполнителей, обеспечивая возможность перенести абстрактные построения, выполненные с помощью компьютера, в предметный мир, получить непосредственное осязательное восприятие исполнителя. Многие дети к началу системных занятий по Лего-конструированию знакомы с понятием «исполнитель», работали или работают с разнообразными исполнителями программно-методической системы «Роботландия», имеют представление о Черепашке – главном персонаже учебной высокоуровневой языковой системы программирования Лого. Тогда первые занятия по программированию, которые обычно сопряжены с трудностями в усвоении детьми материала, проходят легко и эмоционально [5].

Курс робототехники полезно сочетать с обучением алгоритмике. В этом случае многие общие понятия информатики усваиваются ребятами стабильнее, а знания и навыки закрепляются в процессе практической самостоятельной работы.

Учебный курс «Инженерная механика и робототехника» создан и прошел апробацию на базе Дворца творчества «Марьино» системы дополнительного образования детей. Приглашаются к обучению на курсе ученики 3–6 классов. Учебные группы объединяют по 6–8 детей в соответствии с их возрастом и уровнем подготовки. Продолжительность занятий в каждой группе – 2 часа по 2 раза в неделю (144 часа в год в соответствии с учебно-тематическими планами курса). Полный курс составляет два года обучения.

Целью курса является знакомство учащихся с технологией автоматизированного управления, формирование базовых знаний по информатике, алгоритмизации, программированию, отдельным разделам физики, навыков проектной деятельности; развитие конструкторских способностей и активизация системного мышления, которое во многом определяется способностью оперативно обрабатывать информацию и принимать на ее основе аргументированные решения.

В процессе обучения дети знакомятся с устройством и физическими принципами работы современных транспортных средств, строительных механизмов и автоматизированных систем промышленного производства. Имеют возможность воспроизводить прототипы механизмов, исполняющих определенные функции, осваивать навыки работы с механизмами и электроприборами, заниматься программированием.

Важным системным компонентом учебно-воспитательного процесса является проектный метод, который используется совместно с предметным системным обучением, органично дополняя и обогащая его. Использование проектных методик позволяет строить наш курс на принципах проблемного и деятельностного подходов в образовании, личностно ориентированного обучения и коллективного взаимодействия.

В ходе работы над своими моделями ученики вполне самостоятельно актуализируют знания, применяя их практически в незнакомой ситуации, приобретают опыт проектного подхода к решению жизненно важных проблем, который не предусмотрен никаким учебным процессом. Этот опыт заключается в овладении, помимо предметных и житейских умений и навыков, «багажом», связанным непосредственно с проектной деятельностью, – это специфические умения и навыки, такие как проблематизация, целеполагание, организация и планирование деятельности, самоанализ и рефлексия, презентация, коммуникативность, умение принимать решения.

Изменяется традиционная роль учителя в образовательной среде – преподаватель выступает часто в роли консультанта, координатора, эксперта, а отношения «учитель-ученик» носят скорее характер делового сотрудничества. Участвуя в коллективном проекте, ученики имеют возможность реализовать в полной мере свои способности и возможности, а также эффективно действовать в социуме, проявляя выдержку и терпимость к другим. Каждый проект – своеобразная форма повторения или обобщения пройденного материала, каждый очередной проект – ступень для разработки и реализации следующего еще более интересного замысла.

Реализуемая модель требует наличия определенных аппаратных средств и программного обеспечения. Компания Lego выпускает наборы серии Mindstorms для конструирования и программирования роботов начиная с 1999 года, а в российские школы такие наборы стали активно поступать начиная с 2005 года. На сегодняшний день большое количество учебных заведений нашей страны оборудовано образовательными наборами Lego для занятий по робототехнике. В основе каждого такого набора лежит программируемый блок Mindstorms, управляющий работой подключенных к нему моторов и получающий данные с разнообразных датчиков [3, 8].

Первое поколение этих наборов – RCX – выпускалось с 1999 по 2005 годы. Блок Mindstorms RCX позволяет подключить 3 мотора и 3 датчика, среди которых – кнопка, датчик освещенности и датчик температуры. Программируемый блок способен сохранить до 5 программ, память устройства позволяла исполняемой программе объявить не более 32 переменных целочисленного типа и не более 7 подпрограмм/функций. Поддерживается параллельное многопоточное выполнение нескольких подпрограмм. Блок оборудован инфракрасным передатчиком для обмена данными с другими аналогичными блоками посредством передачи сообщений.

Более новая версия устройства – NXT – выпускается с 2006 года по настоящее время. Программируемый блок NXT не имеет таких жестких ограничений на исполняемые программы, позволяет работать с файлами, воспроизводить звуки, выводить текст и изображения на экран. Система также позволяет работать с массивами, строками и структурами данных. Число портов для датчиков увеличено до 4, появляются датчики звука, ультразвуковой дальномер, компас, акселерометр. Моторы оснащены счетчиками угла поворота. Для передачи данных вместо инфракрасного порта используется Bluetooth, но также возможно подключение модуля инфракрасной связи для обмена данными с блоками RCX [2, 8].

Помимо электроники, наборы содержат большое количество деталей (600–800) для сборки различных механизмов. При том, что в сравнении с другими наборами Lego это достаточно внушительный запас, такого количества деталей хватит только для простейших учебных задач. Для расширения элементной базы существуют специальные наборы Lego education, содержащие большое число строительных элементов, соединений и шестеренок. Другой источник деталей – обычные модели Lego technic, продающиеся в магазинах игрушек. В отличие от образовательных наборов, такие конструкции содержат меньшее количество технических элементов и шестеренок, но в них есть декоративные элементы, а также специализированные детали для реализации функционала данной конкретной модели. Большинство учебных заведений, оборудованных наборами Mindstorms, имеют в своем распоряжении также и ресурсные наборы Lego, в противном случае расширение элементной базы за счет обычных моделей не составляет проблемы [3].

Достаточно распространенной является ситуация, когда для реализации собственных творческих проектов команде учащихся может не хватить функциональности одного комплекта Mindstorms в связи со сложностью поставленной ими задачи. В связи с этим оптимальное аппаратное оснащение курса формируется из расчета 2 комплекта Mindstorms на каждую команду, готовящую творческий проект, плюс 1–2 комплекта для общих учебных задач.

В то время, как вопрос выбора аппаратного оснащения для организации курса робототехники не вызывает особой проблемы, выбор программного обеспечения представляет собой непростую задачу. Существует большое количество инструментов для программирования блоков Lego Mindstorms различной сложности, каждый из которых имеет свои преимущества, поэтому выбор среды разработки во многом зависит от задач конкретного курса и возраста учащихся.

Обзор сред программирования блоков Mindstorms

1) Графические среды, входящие в состав наборов Mindstorms. Это наиболее простые варианты среды программирования для роботов, тем не менее их возможностей достаточно для реализации любых учебных задач, не требующих сложных математических вычислений. Каждая команда представлена в виде пиктограммы, которая перетаскивается из библиотеки в рабочую область на «временную линию». Дизайн элементов интерфейса среды программирования NXT-G (графическая среда, поставляемая в комплекте наборов Mindstorms NXT) максимально приближен к дизайну элементов Lego, таким образом, создается впечатление, что программа собирается из деталей аналогично конструктору. Это существенное достоинство инструмента на начальных этапах обучения детей программированию. Недостаток таких систем – они имеют существенные недоработки, с ростом сложности про-

грамм инструменты начинают работать нестабильно: возникают ошибки, зависания и потери участков программ, что пугает детей, и им становится сложно работать. Другой недостаток – неоптимальная компиляция программ. Созданная в такой среде программа занимает на устройстве в среднем в 6 раз больше памяти, чем аналогичные программы на языке более низкого уровня. Все это позволяет сделать вывод, что подобные среды подходят только для простейших учебных задач и являются для них оптимальным выбором [6].

2) LabVIEW. Вышеозначенные учебные среды программирования были разработаны Массачусетским Технологическим институтом на основе более серьезного инструмента – графической среды LabVIEW, созданной для инженеров электронных устройств от компании National Instruments. Этот инструмент позволяет создавать программы, ориентированные на конкретные устройства или системы устройств, он содержит огромный набор средств для сборки программ для практически всех современных устройств, а также для удобного построения схем взаимодействия устройств. На сегодняшний день этот инструмент очень популярен. По сути, он – сильно продвинутый вариант простейших сред Mindstorms, более стабильный, позволяющий выполнять сложные математические операции, а также легко организовывать взаимодействие между блоком NXT и компьютером, либо между несколькими блоками. Последнее является существенным плюсом LabVIEW, у других инструментов передача данных организована сложнее. Переход от простейших сред программирования к LabView гораздо легче для школьников, чем переход от графической среды к текстовой. Основным недостатком LabVIEW является его ограниченная доступность. Другой минус инструмента состоит в том, что с ростом сложности программы, нарисованной в виде пиктограмм, теряется наглядность, и если простая программа выглядит понятно, то более сложная – хаотично, большое количество пиктограмм на экране одновременно сильно затрудняет ее читаемость [4].

3) Текстовые среды программирования с использованием языков типа C. Наиболее распространенный инструмент – BrickCC, в котором используется C-подобный язык NXC. Это свободно распространяемая программа, имеющая большое число различных инструментов для работы с блоками Lego Mindstorms, фактически может полностью заменить стандартное программное обеспечение Lego (кроме драйверов). Встроенные библиотеки языка позволяют работать с устройством на различных уровнях, присутствуют низкоуровневые средства обращения к входам и выходам устройства, обращение к физическим адресам памяти RCX, а также высокоуровневые команды управления моторами и получения данных с датчиков, достаточно простые для использования детьми.

Классическая задача – передвижение робота вдоль черной линии на полу с использованием двух датчиков освещенности – на языке NXC может выглядеть, например, так:

```
int right, left;
task main()
{
  SetSensorLight(S1, true); // включаем датчик света S1
  SetSensorLight(S2, true);
  right = SENSOR_1; // получаем данные с датчика
  left = SENSOR_2;
  OnFwd(OUT_BC, 100);
  while(right >= 40 || left >= 40)
  {
    right = SENSOR_1;
    left = SENSOR_2;
    if (right < 40) // правая сторона на черном, мы отклонились влево
    { // повернем вправо
      Off(OUT_B);
      OnFwd(OUT_C, 50);
    }
    else if (left < 40) // отклонились вправо
    { // повернем влево
      Off(OUT_C);
      OnFwd(OUT_B, 50);
    }
  }
}
```

```

    }
    else
    {
        OnFwd(OUT_BC, 100);
    }
}
Off(OUT_BC);
}

```

Особо стоит отметить удобный механизм шаблонов – заранее может быть создан текстовый файл шаблона, в котором можно перечислить любые строки, а затем при редактировании программы нажатие правой кнопкой мыши на строке открывает этот список строк, после чего любую строку данного файла можно вставить в текст программы нажатием мыши. Таким образом, можно обернуть комментариями на русском языке все основные функции языка, чтобы облегчить понимание для детей, незнакомого с английским языком. Синтаксис комментариев аналогичен комментариям языка C: начинающая с /* и заканчивая */. Текстовые строки записываются в кавычках, например, "строка". Простейшие примеры строк шаблона:

- OnFwd(OUT_A, 100); /* Включи мотор A */
- while ("тут условие") /* повторяй, пока условие верно */

В приведенных шаблонах \= означает переход на новую строку, \> – переход на одну табуляцию, а \> – переход на новую строку с уменьшением отступа. Последний шаблон при вставке в текст программы будет выглядеть следующим образом:

```

while ("тут условие") /* повторяй, пока условие верно */
{
    "повторять это"
}

```

4) Программа на C++ или аналогичном языке, исполняющаяся на компьютере и отправляющая роботу управляющие сигналы по беспроводной связи. Для учебных целей в нынешней реализации это неприменимо, но, в принципе, такие средства (библиотеки для языков C++, C#, Java) существуют. В перспективе они могут быть модифицированы для учебных целей, например, на их основе могут быть разработаны учебные средства для управления роботом в реальном времени. Возможно, с помощью этих библиотек будут написаны простые учебные программы, наподобие ПиктоМира, которые бы позволяли запускать программу для робота без компиляции и загрузки на устройство.

При построении курса использовались методические пособия, разработанные образовательным подразделением компании Lego, материалы Института Новых Технологий (ИНТ) и Московского Института Открытого Образования (МИОО), а также книги «Программируем микроконтроллер NXT в LabVIEW» Л. Г. Беливской и А. Е. Беливского и «Робототехника для детей и родителей» С. А. Филиппова. Указанные пособия рекомендуются к прочтению при организации курсов робототехники на базе конструктора Lego [2, 4, 7].

Для нашего курса были выбраны среды программирования NXT-G – для первого года обучения и VixCC – для второго. Курс предполагает изучение основных механизмов программного управления роботом на основе графической среды NXT-G как наиболее простого и доступного для детей инструмента, для которого легко проводить параллели с другими средами графического программирования, уже знакомыми учащимся (например, Scratch). На второй год обучения, либо, при быстром усвоении материала, к концу первого года предполагается постепенный переход к низкоуровневой системе текстового программирования на базе языка C. Мы сознательно отказываемся от использования среды LabVIEW при создании курса. Это связано с тем, что одной из важнейших его задач мы видим организацию перехода детей от графического программирования к текстовому. VixCC является уникальным инструментом разработки, сочетающим в себе наглядность исполнителя программ и текстовую среду. Для выполнения простых учебных задач и проектной деятельности младших школьников вполне достаточно стандартных сред Lego, более того, при изучении основ работы с программированием роботов гораздо естественнее воспринимается графическая среда, выполненная в дизайне конструктора Lego. С повышением сложности задач рекомендуется сразу смотреть в сторону текстовых сред разработки, не останавливаясь на LabVIEW.

Учитывая возраст учащихся, предполагается незнание ими английского языка, что делает необходимым использование механизма шаблонов с русскоязычными комментариями в среде VbixCC. Следует подчеркнуть, что сам по себе язык программирования NXC в среде VbixCC в чистом виде будет достаточно сложен для младших школьников. Задача преподавателя – грамотно организовать систему шаблонов, подобрать хорошие комментарии, исходя из того, чему дети обучались ранее, чтобы текстовые команды языка не были для них чем-то непонятным, а смотрелись естественно.

В соответствии с поставленными задачами курс предлагает для изучения следующие темы:

1. Общие представления об архитектуре ЭВМ.
2. Понятие о данных и переменной.
3. Последовательное программирование:
 - Основные управляющие алгоритмические структуры (циклы, условия).
 - Структура программы, подпрограммы. (Последовательное программирование.)
4. *Более детальное понятие об устройстве памяти. Типы переменных, адресация памяти (для старших).
5. Особенности построения программ, работающих в бесконечном цикле (реактивное программирование, черный ящик). Понятия состояния и перехода. *Пример реализации.

Двухгодичный курс поддерживает концепцию непрерывного информатического образования, где принцип дидактической спирали является одним из факторов структуризации. В разных частях дисциплины предполагается возврат к уже усвоенным понятиям, но на новом уровне усложнения и обобщения изучаемого материала. Так, названные темы, кроме отмеченных звездочкой, являются общими для первого и второго года обучения, но обсуждаются с учениками по-разному, с учетом уровня освоения ими информации. Вопрос о более детальном устройстве памяти и реализации программ, работающих в бесконечном цикле, на языке NXC предлагаются к изучению для второго года.

Архитектура ЭВМ

Многие авторы курсов по робототехнике избегают упоминания об архитектуре ЭВМ в основном из-за ограниченности времени курса. Однако именно модели роботов позволяют наиболее наглядно вести разговор об устройстве вычислительных систем. Программируемый блок NXT содержит набор простейших тестовых программ TRY ME, которые демонстрируют работу всех устройств, подключенных к блоку. Примеры – при нажатии кнопки вращается мотор, либо при изменении освещения в помещении изменяется изображение на экране. С помощью одного только блока NXT и сопутствующей электроники, даже без сборки модели, можно сформировать понятия внешних устройств ввода, вывода, памяти, процессора. Учитывая тенденции развития современного информационного общества, естественно предположить, что многим детям легко понять назначение всех внешних устройств, подключаемых к блоку Mindstorms, основной задачей данного раздела является систематизация этих понятий.

Основные команды языка программирования. Управление внешними устройствами

Учитывая потребность детей в осознании результатов своей работы, естественно, что первыми командами языка программирования будут команды управления внешними устройствами, в первую очередь – моторами, для того чтобы заставить робота двигаться. Для простейшей линейной программы достаточно всего 2 команды языка NXT-G – команды управления мотором и команды «ждать определенное время». В среде VbixCC это будет уже 4 команды (3 команды управления моторами – OnFwd, OnRev, Off и команда ожидания Wait).

Если с начальным шагом в программировании все источники совершенно солидарны, то далее в различных методических пособиях по организации курсов начинаются различия. Одни авторы предлагают изучать отдельные датчики, постепенно подводя учеников к понятиям логических управляющих структур, другие – строить и изучать классические «хрестоматийные» модели, например, робота для езды вдоль линии. В любом случае для дальнейшего продвижения предстоит разговор об управляющих конструкциях и о данных и переменных. Поскольку об организации памяти и внешних устройств говорилось в разделе про архитектуру ЭВМ, было бы естественно дальше рассказать о переменных как о средстве хранения всех тех данных, которые приходят от внешних носителей информации.

Логичным завершением разговора о переменных будет демонстрация их применения в различных ситуациях. Очень полезно использование команды вывода значения переменной на экран блока NXT, на начальных этапах это облегчит понимание сути переменной, а на более поздних – полезно для отладки программ.

Хорошим решением также будет описать свой набор подпрограмм и использовать его вместо основных команд языка. Например, описать новую команду «запись данных в переменную», которая, помимо записи данных, еще и отобразит ее на экране блока Mindstorms, либо подпрограмму для организации циклов со встроенными задержками и звуковыми сигналами, чтобы наглядно продемонстрировать работу цикла. Возможно также использование подпрограммы, выполняющейся параллельно основной программе, которая в бесконечном цикле считывает показания всех датчиков и записывает в переменные.

Говорить о структуре программ можно в том случае, когда все ранее введенные понятия хорошо усвоились. В этот момент учащиеся уже должны начинать работу над собственными проектами. Мы предполагаем, что все творческие проекты выполняются детьми в командах, поэтому будет естественно, если каждый из участников проекта будет ответственен за свою часть программы. Это и будет являться поводом к тому, чтобы заговорить о разделении программ на подпрограммы – функции. В этот же момент можно начать организацию перехода от среды NXT-G к VixCC для тех детей, которые хорошо усвоили весь предыдущий материал. При грамотной организации механизма шаблонов учащиеся смогут формировать текстовые программы с помощью мыши так же, как и в графической среде.

VixCC дает возможность записывать данные в память блока RCX по физическому адресу памяти. Это позволяет наглядно продемонстрировать архитектуру памяти для старших учащихся при более детальном разговоре об архитектуре ЭВМ.

Особенности построения программ, работающих в бесконечном цикле

Предполагается, что последний пункт программы является наиболее важным, так как в нем содржится кульминация всех полученных детьми знаний, а также понятие о том, как работают реальные современные программы. Именно такой тип программ детям, скорее всего, предстоит реализовать для своих творческих или конкурсных проектов. Робот представляет собой замкнутую систему, «черный ящик», который принимает данные извне и превращает их в некоторые конкретные выходные действия. На этом этапе возможно рассмотрение таких классических задач школьной робототехники, как движение вдоль линии или сортировка деталей по цветам. Практически все подобные задачи, равно как и программы, которыми пользуются люди в повседневной жизни, требуют наличия бесконечного цикла и реакций на события.

Наблюдения и рекомендации по организации курса

Важным фактором в работе с детьми в рамках курса, построенного вокруг конструктора Lego, является то, что детям нравится играть в конструктор и не нравится теория. Школьники, в особенности младшего возраста, с удовольствием будут собирать модели даже по инструкции. В работе с конструктором заключается основа мотивации учащихся, и поначалу, когда они еще не приступили к работе над собственным проектом, они не будут испытывать желания изучать программирование. Эта проблема поднималась в публикациях зарубежных педагогов, которые формулируют ее следующим образом: детям не нужно программировать робота, чтобы заставить его работать, ведь в их воображении он прекрасно функционирует и без всяких программ и сложной теории. Эффективность курса информатики будет существенно ниже, если учащимся не показать реальную необходимость программирования. Из этого следует, что при организации курса робототехники нужно планировать программистскую и инженерную части так, чтобы они были тесно связаны. Модели должны строиться с таким расчетом, чтобы в определенный момент была возможность постановки очевидной для детей задачи, которую они могли бы решить, будучи верно запрограммированными. Один из авторов, поднимавших указанную проблему, описывал свой курс, где дети постепенно строили робота, который шел по определенному, придуманному им сюжету, преодолевал преграды на своем пути с помощью также постепенно усложняющихся программ. В начале занятия он рассказывал детям новую главу сюжета, что сделает их робот, например, переберется через реку или победит врага, затем в течение занятия выполняется модификация робота, позволяющая ему выполнить поставленную задачу, и пи-

шется программа. В конце занятия дети запускают робота, он выполняет задачу и получает приз. В качестве приза выступает новая команда языка программирования, которая будет использована на следующем занятии. Основной вывод из этого – на каждом занятии, на котором изучается теоретический материал, детям следует дать понять, какой эффект они смогут получить после его освоения, а в конце занятия они должны иметь возможность увидеть наглядный результат своей работы, полученный с помощью приобретенных знаний [3].

Еще один подход к решению этой проблемы основан на примере позитивного лидера и потребности детей в успешной деятельности. Старшие учащиеся могут демонстрировать свои модели младшим, побуждая их работать над собственными конструкциями для аналогичной демонстрации в будущем. Наблюдая за тем, как функционируют другие модели, осознавая требования, предъявляемые к ним, происходит личное присвоение детьми проблемы, усиливается мотивация к ее решению.

Также необходимо грамотно комбинировать занятия по конструированию и по программированию. В силу ограниченности времени зачастую бывает очень сложно в рамках одного урока уделить внимание и инженерной составляющей курса, построить интересную и функциональную модель или добавить некоторое решение к уже имеющейся алгоритмической составляющей. В связи с этим распространенной ситуацией является то, что некоторые занятия будут посвящены исключительно конструированию, а другие – только программированию. Эти ситуации сопряжены с рядом трудностей. С одной стороны, если программистский аспект будет слишком большим, у детей пропадет мотивация, они будут пропускать занятия, проявлять меньше активности. С другой стороны, если будет больше конструкторской части, дети не будут усваивать важные знания по программированию, забывая все ранее пройденное. Не рекомендуется посвящать больше 2 занятий подряд программированию без работы с конструктором и больше 3 занятий подряд без активного использования полученных знаний по программированию. При этом даже если занятие планируется как исключительно инженерное, повторение пройденного теоретического материала бывает весьма полезным. Тот факт, что быстрое повторение материала позволит скорее приступить к работе над моделями, является хорошим мотивирующим фактором для изучения теории.

В ходе реальной работы над моделями наиболее сложным был вопрос о степени самостоятельности учащихся, которая в каждом конкретном случае зависела от возрастных и индивидуальных особенностей детей, от их предыдущего опыта Лего-конструирования, от характера отношений в группе и многого другого. При этом важно было избежать перегибов как в ту, так и в другую сторону.

Многие преподаватели, имеющие опыт ведения занятий по робототехнике на базе конструктора Lego, отмечают следующий факт: когда дело доходит до конструктора, дети хотят делать все сами, не слушая руководителя. Наблюдения показывают, что такой фактор действительно присутствует, и его природа несколько сложнее. Причина этого отчасти заключается в том, что преподаватели, ведущие подобные кружки в большинстве учебных заведений, не имеют большого опыта в Лего-конструировании, а отчасти – в том, что у них отсутствует заинтересованность в процессе построения учениками их творческих работ. Указанные факты приводят к тому, что преподаватель не проявляет большого внимания к работе учащихся, и тем самым не создается среда для эффективного взаимодействия учителя и ученика, формирование которой является, как было подчеркнуто выше, одной из важнейших задач. Для детей естественной становится ситуация, когда они собирают конструктор исключительно сами или со своими ровесниками, а взрослые проявляют мало интереса к Lego. Как следствие, теряется авторитет учителя в глазах учеников, появляется уверенность в том, что они более компетентны в вопросах работы с конструктором Lego. Если подобная ошибка будет допущена, курс робототехники не будет приносить никаких положительных результатов. Из этого можно сделать вывод, что крайне важно принимать вместе с детьми активное участие в том, что им больше всего интересно, только тогда взаимодействие учителя и ученика будет эффективным.

Утверждая это, следует заметить, что для человека, не имеющего опыта сборки технических Lego-моделей, будет несколько затруднительно выполнить подобную рекомендацию. Наилучший выход из данной проблемы – более тесное знакомство с возможностями наборов Lego Technic и Mindstorms. В частности, это сборка по инструкции нескольких сложных Lego-наборов, исследование закономерностей в конструкции этих моделей, принципов соединения деталей, способов обеспечения прочности конструкции, легкости вращения и движения всех узлов. Также полезными являются видеоролики любителей-

энтузиастов Lego Mindstorms из разных стран мира, с демонстрацией возможностей их моделей. Подобные видео могут быть найдены на сайте <http://youtube.com> по ключевому слову Mindstorms.

На фоне проблемы, связанной с недостатком внимания по отношению к творческому процессу, проявляется еще один важный аспект организации курса, который необходимо принять во внимание. Он связан с тем, что дети не знают границ возможностей конструктора, а все элементы Lego имеют вполне определенный предел прочности, и особенно это касается сложных технических деталей, и в еще большей степени – электроники. Поэтому крайне важно привить детям аккуратность и культуру работы с такими сложными элементами. Неосторожность и плохо рассчитанная конструкция чаще всего приводят к поломке деталей. Учитывая нагрузку на них в сложных механизмах, сломанные технические детали невозобновимы. Плохой редуктор, большая масса и трение в соединениях, ограниченный угол поворота детали могут уничтожить либо деталь (в случае мощного мотора), либо сам мотор. Зачастую детям это совершенно неочевидно, потому что до этого они собирали по инструкциям модели, грамотно продуманные инженерами компании Lego, где возможность поломки исключена и все механизмы работают без проблем. Следует отметить, что даже инженеры компании Lego допускают ошибки, в частности, из-за одной такой ошибки в одной из моделей 2010 года выпуска перегорали моторы, вследствие чего модель была снята с продажи.

В особенности подобная проблема относится к программируемым моделям. Если в конструкции, управляемой руками, один из узлов собран неверно, это может быть замечено относительно легко, поскольку трудность вращения можно почувствовать. Если же существует ошибка в программируемой модели или, что еще неприятнее, ошибка в программе, то человек может сразу не заметить проблему, пока не станет слишком поздно. Модель будет поставлена на стол, программа запущена, и, если она начинает ломать модель, можно не успеть ее вовремя остановить. Подобные ситуации случаются на занятиях робототехники с использованием Lego Mindstorms постоянно, поэтому нужно внимательно контролировать все, что делают учащиеся, проверять написанные ими программы, собранные механизмы на предмет ошибок. А также, что более важно, у детей должна сформироваться культура самостоятельного контроля качества своих построек.

Заключение

Двухгодичный курс «Инженерная механика и робототехника» представляет собой сложившуюся образовательную систему, которая занимает свое определенное место в структуре информатического образования детей. Обучаясь на этом курсе, ученики получают системные знания, формируют умения, навыки, компетенции в соответствии со своими возрастными и личностными особенностями. Программа курса поддерживает концепцию формирования операционного стиля мышления у школьников, функционального, деятельностного обучения, формирования исследовательских навыков, активного творчества и успешного группового взаимодействия. Возможность реализации этих концепций обеспечивается включением проектных методик в учебно-воспитательный процесс, использованием проектного метода не вместо традиционного предметного обучения, а вместе с ним, органично дополняя и обогащая его. Важным представляется наличие комплексного подхода к разработке методик, содержанию обучения, выбору аппаратного и программного обеспечения курса. Очевидно, что такая программно-методическая система может быть наиболее успешно реализована в условиях дополнительного образования, где есть возможность гибкого подхода к распределению учебного времени, неформальной организации учебного процесса и наличия небольших учебных групп.

Библиографический список

1. Белиовская, Л. Г., Белиовский, А. Е. Программируем микроконтроллер NXT в LabView [Текст] / Л. Г. Белиовская, А. Е. Белиовский. – М. : ДМК Пресс, 2010.
2. Злаказов, А. С. Уроки Лего-конструирования в школе [Текст] / А. С. Злаказов, Г. А. Горшков, С. Г. Шевалдина. – М. : БИНОМ, 2011.
3. Курсы по робототехнике от МИОО [Электронный ресурс]. – Режим доступа : <http://learning.9151394.ru/course/category.php?id=256>.
4. Паперт, С. Переворот в сознании : дети, компьютеры и плодотворные идеи [Текст] / С. Паперт. – М. : Педагогика, 1989.
5. Первин, Ю. А. Роботландия [Текст] / Ю. А. Первин [и др.]. – М. 1991.
6. Филлипов, С. А. Робототехника для детей и родителей [Текст] / С. А. Филлипов. – СПб. : Наука, 2010.
7. <http://doublebrick.ru>.
8. <http://wikipedia.org>.

Bibliograficheskiy spisok

1. Beliovskaya, L. G., Beliovskiy, A. Ye. Programmiruyem mikrokontroller NXT v LabView [Tekst] / L. G. Beliovskaya, A. Ye. Beliovskiy. – M. : DMK Press, 2010.
2. Zlakazov, A. S. Uroki Lego-konstruirovaniya v shkole [Tekst] / A. S. Zlakazov, G. A. Gorshkov, S. G. Shevaldina. – M. : BINOM, 2011.
3. Kursy' po robototekhnike ot MIOO [Elektronnyy resurs]. – Rezhim dostupa : <http://learning.9151394.ru/course/category.php?id=256>.
4. Papert, S. Perevorot v soznanii : deti, komp'yutery' i plodotvornyye idei [Tekst] / S. Papert. – M. : Pedagogika, 1989.
5. Pervin, Yu. A. Robotlandiya [Tekst] / Yu. A. Pervin [i dr.]. – M. 1991.
6. Fillipov, S. A. Robototekhnika dlya detey i roditeley [Tekst] / S. A. Fillipov. – SPb. : Nauka, 2010.
7. <http://doublebrick.ru>.
8. <http://wikipedia.org>.