Научно-исследовательская работа

Обучение учащихся основам программирования

с использованием среды Robotc

Выполнили: ст. гр. 18ФПИ1 Шашлова Алина Алексеевна,   
 Гудорожникова Елена Алексеевна

Руководитель: Кочеткова Ольга Анатольевна

Содержание

[Введение 3](#_Toc448405598)

[Глава 1 Теоретические основы разработки курса «Робототехника. Программирование» 6](#_Toc448405599)

[1.1 Внеклассная работа по информатике, её особенности 6](#_Toc448405600)

[1.2 Содержание линии «Алгоритмизация и программирование» в школьном курсе информатики и ИКТ 14](#_Toc448405601)

[1.3 Изучение робототехники в школе 19](#_Toc448405602)

[Глава 2 Обучение учащихся основам программирования с использованием среды Robotc 28](#_Toc448405604)

[2.1 Определение структуры и содержания курса «Робототехника. Программирование» 28](#_Toc448405605)

[2.2 Тематическое планирование факультативного курса 34](#_Toc448405606)

[2.3 Конспекты факультативных занятий по курсу «Робототехника. Программирование» 42](#_Toc448405607)

[Заключение 57](#_Toc448405610)

[СПИСОК ЛИТЕРАТУРЫ 59](#_Toc448405611)

[Приложение 61](#_Toc448405612)

# **Введение**

Федеральный государственный образовательный стандарт (ФГОС) основного общего образования поколения, учитывая развитие современных технических средств обучения, вносит изменения в содержание современного образования учащихся. Эти изменения неизбежно связаны с требованиями к материально-техническим условиям реализации основной образовательной программы, одним из которых является обеспечение возможности «проектирования и конструирования, в том числе моделей с цифровым управлением и обратной связью, с использованием конструкторов; управления объектами; программирования» [1].

В рамках уроков информатики эти требования, по нашему мнению, могут быть реализованы с помощью внедрения особой образовательной технологии – образовательной робототехники.

Робототехника – прикладная наука, занимающаяся разработкой автоматизированных технических систем. Она опирается на такие дисциплины, как электроника, механика, программирование. Соответственно в школе робототехника может интегрироваться с такими предметами как математика, физика, информатика и ИКТ [7].

В первую очередь занятия по робототехнике рассчитаны на общенаучную подготовку учащихся, развитие их мышления, логики, математических и алгоритмических способностей, исследовательских навыков. Кроме того, учащиеся, прошедшие обучение, могут продолжить занятия с учителями естественнонаучных предметов, которые смогут использовать их знания оборудования и языка программирования для проведения научно-исследовательских работ с использованием самых современных средств сбора и обработки данных.

Актуальность работы заключается в том, что направление «Робототехника. Программирование» является относительно новым для школ, поэтому наблюдается дефицит в учебно-методическом обеспечении проведений занятий по робототехнике. Кроме того, как правило, занятия по робототехнике сводятся только к конструированию роботов, а программирование остается вне поля зрения учеников и учителей. Поэтому возникает необходимость в факультативным курсе, сочетающий в себе как элементы конструирования, так и элементы программирования роботов. Существует несколько сред программирования, используемых для разработки программы управления компьютером NXT: NXT-G, Robolab, **TrikStudio, Arduino**, RobotC и др. Проведя их анализ, мы сделали вывод, что среда RobotC наиболее подходим для обучения учащихся программированию, а именно, на языке С.

Цель работы – разработка и учебно-методическое сопровождение курса «Робототехника. Программирование» для учащихся средней школы.

Объектом исследования является учебно-воспитательный процесс, реализуемый на факультативных занятиях в среднем звене образовательных учреждений.

Предметом исследования является структура, содержание и методы проведения факультативных занятий в среднем звене образовательных учреждений.

Задачи работы:

1. Изучить методическую и научную литературу по теме исследования.
2. Провести анализ учебников информатики средней школы на предмет изучения линии «Алгоритмизация и программирование».
3. Определить место курса «Робототехника. Программирование» в образовательном пространстве школы.
4. Определить структуру и содержания факультативного курса «Робототехника и программирование».
5. Разработать методическое обеспечение факультативного курса.

Практическая значимость заключается в разработке структуры и содержания факультативного курса «Робототехника. Программирование», который может быть использован учителями средних общеобразовательных учреждений и учреждений дополнительного образования в своей педагогической деятельности, студентами вузов при прохождении педагогических практик.

# **Глава 1 Теоретические основы разработки курса «Робототехника. Программирование»**

## **1.1 Внеклассная работа по информатике, её особенности**

Сегодня для образовательного учреждения на первое место выходит вопрос организации внеурочной деятельности учащихся, которые должны быть вовлечены в исследовательские проекты, творческие занятия, спортивные мероприятия, в ходе которых научатся изобретать, понимать и осваивать новое, быть способными выражать собственные мысли, уметь принимать решения, формулировать интересы и осознавать возможности. Такая возможность предоставляется Федеральным государственным образовательным стандартом нового поколения.

Внеурочная познавательная деятельность учащихся может быть организована в форме факультативов, кружков познавательной направленности, проектной деятельности и т. д.

Внеклассная работа – различные воспитательно-образовательные мероприятия, выходящие за рамки обязательных учебных программ и проводимые школой во внеурочное время [7].

В задачи внеклассной работы по информатике входит:

1. Углубление знаний учащихся теоретических основ информатики, программирования, изучение архитектуры ЭВМ и сетей, работа с программным обеспечением (MS Word, MS Excel, Ms PowerPoint, графические редакторы и др.).
2. Формирование у учащихся навыков работы с ПК и различным программным обеспечением, интереса к исследовательской работе.
3. Воспитание интереса к чтению как обычной, так и электронной научной литературы, формированию умений и навыков в работе с ними.
4. Профориентационная работа с учащимися.

Перечислим особенности внеклассной работы по информатике:

1. По своему содержанию она строго не регламентирована государственной программой. Но содержание учебного материала соответствует знаниям и умениям учащихся, т.е. их уровню подготовки.
2. Нет четкого регламента по времени, т.е. внеклассные занятия в зависимости от содержания и формы проведения могут быть рассчитаны и на 5 – 10 минут (занимательные переменки), и на 1,5 – 2 часа.
3. Для внеклассной работы учащиеся данной школы могут объединяться в группы, обучаясь, либо в одном и том же классе, либо в разных классах.
4. Внеклассная работа характеризуется многообразием форм и видов: групповые занятия, викторины, вечера, олимпиады, кружки, дистанционные формы обучения и др.
5. Занимательность предлагаемого материала либо по содержанию, либо по форме, широкое использование игровых форм проведения занятий и элементов соревнования на них.
6. Оснащенность кабинета информатики соответствующим техническим оборудованием: наличие достаточного количество компьютеров и периферийных устройств (принтеров, сканеров, техническую проектор и т.д.); наличие Интернета и научной литературы.

Внеклассная работа проводится на основе общих принципов. Рассмотрим их [10].

Принцип целевой установки требует точного планирования учебно-воспитательных задач, оценки достигнутых результатов, всестороннего анализа проведенной работы. Перед началом необходимо определить ожидаемые результаты. Это помогает сформулировать задачи таким образом, чтобы они способствовали достижению общей цели. Т.о., необходимо наличие четкой формулировки цели и воспитательных задач каждого намечаемого мероприятия (занятия).

Сущность принципа связи внеклассной работы с жизнью и практикой заключается в том, внеклассная работа должна иметь общественно полезную направленность.

Принцип воспитания в коллективе и через коллектив следует непосредственно из целей обучения и воспитания. Внеклассная работа открывает широкие возможности для постоянного участия в различной коллективной деятельности, которую следует организовывать с учетом законов формирования коллектива и уже достигнутого уровня его развития.

Принцип единства и целостности учебно-воспитательного процесса означает органичное соединение учебной и внеклассной работы. Неразрывность этих сторон педагогического процесса при учете индивидуальных особенностей учащихся – важное условие повышения его эффективности. Этот принцип требует и преемственности между возрастными ступенями учебно-воспитательного процесса (младшими, средними и старшими классами), а также учета уровня развития и подготовленности учащихся.

Принцип учета возрастных и индивидуальных особенностей во внеклассной работе означает, что она должна строиться с опорой на силы и возможности, присущие данному возрасту, и содействовать дальнейшему развитию каждого учащегося.

Содержание внеклассной работы по информатике в начальных классах должно быть рассчитано на расширение кругозора учащихся. Внеклассную работу с учащимися среднего школьного возраста следует строить с учетом уже приобретенных ими знаний, накопленного жизненного опыта. Задача учителя – помочь каждому ученику найти такое внеклассное занятие, которое смогло бы удовлетворить его интересы и содействовало бы развитию его способностей.

Старшеклассники стремятся и интересуются серьезными занятиями, выходящими за рамки учебных программ. В этом возрасте происходит выбор жизненного пути, и потому важна профориентация учащихся во время внеклассной работы.

Принцип единства требовательности и уважения к личности определяет взаимоотношения учителей и учащихся в процессе внеклассной работы, позволяет выявить те увлечения и интересы учащихся, в которых наиболее полно проявляются их положительные качества. Опираясь на них, учитель может добиваться общей перестройки поведения и развития интересов учащихся, улучшения их успеваемости и дисциплинированности.

Выделяют два вида внеклассной работы по информатике [9]:

* 1. Работа с учащимися, отстающими от других в изучении программного материала (дополнительные внеклассные занятия).

В идеальном случае первый вид внеклассной работы должен иметь ярко выраженный индивидуальный характер и проявляться лишь в исключительных случаях (например, в случае продолжительной болезни учащегося, перехода из школы другого типа т. п.).

Однако в настоящее время эта работа требует еще значительного внимания со стороны учителя информатики. Основной целью ее является своевременная ликвидация (и предупреждение) имеющихся у учащихся пробелов в знаниях и умениях по курсу информатики.

* 1. Работа с учащимися, проявляющими к изучению информатики повышенный интерес. Цели таких занятий следующие: развитие устойчивого интереса к информатике; расширение и углубление знаний по программному материалу; развитие способностей учащихся и привитие им определенных навыков научно-исследовательского характера; развитие у учащихся умения самостоятельно и творчески работать с учебной и научно-популярной литературой; расширение и углубление представлений учащихся о практическом значении информатики в жизни общества; расширение и углубление представлений учащихся о культурно-исторической ценности информатики, о роли информатики в мировой науке.

Содержание внеклассных занятий обычно имеет практическое значение и большой познавательный интерес – компьютерные технологии обработки любого вида информации, в частности, мультимедиа, гипертекст, Интернет и др.

В качестве форм организации образовательного процесса в настоящее время используются:

1. Элективный курс – обязательный для посещения курс по выбору учащихся, входящих в состав содержания обучения на старшей ступени школы и реализуемые за счет школьного компонента учебного плана.

2. Кружок – основной и наиболее распространенный вид групповой внеклассной работы. Организуется он из учащихся одного или параллельных классов; постоянный состав обычно не превышает двадцати человек. Основу кружка, как правило, составляют учащиеся, интересующиеся конкретной предметной проблемой. Кружок относится к групповым формам работы с закрепленным составом учащихся.

3. Факультативные занятия – форма учебной работы, состоящая в развитии способностей и интересов учащихся в сочетании с общеобразовательной подготовкой. Факультативные занятия организуются и реализуются в соответствии с интересами и индивидуальными способностями учащихся.

Классификация факультативов:

* + - факультативы по предметам обучения, входящим в учебный план, на которых углубленно изучается содержание учебного предмета, систематизируются и обобщаются полученные знания;
    - прикладные факультативные занятия проводятся с целью знакомства с важнейшими путями и методами применения знаний на практике, развития интереса к современной технике, производству;

– факультативные спецкурсы позволяют углубленно рассмотреть отдельные вопросы изучаемого курса, а также проблемы, выходящие за рамки учебной программы;

* + - межпредметные факультативы помогают ученикам глубже понять окружающий мир, узнать, как применяются знания по различным дисциплинам в определенных отраслях производства.

Сравним элективные и факультативные курсы.

Целью факультативных занятий является углубление знаний, развитие интересов, способностей и склонностей учащихся, их профессиональное самоопределение и т.д. У элективных курсов цели аналогичны, но могут конкретизироваться в зависимости от направленности.

Факультативные и элективные курсы объединяет отсутствие государственных стандартов и государственного итогового контроля по результатам их изучения (например, в форме ЕГЭ или ОГЭ).

Отличием является то, что элективные курсы являются обязательным образовательным компонентом для всех учеников 9–11-х классов общеобразовательных школ, а их факультативные курсы – это необязательные учебные занятия для всех учащихся [14].

Отличительная черта факультативных и элективных курсов – их разная продолжительность. Факультативные курсы представлены программами, рассчитанными на весь учебный год (минимум – 34 ч). Продолжительность элективных курсов профильного обучения может быть более длительной (от 6–8 до 72 ч, могут быть рассчитаны на 1–2 месяца, на четверть, полугодие). Т.о., элективные курсы в отличие от факультативов могут быть краткосрочными.

Факультативные курсы, как правило, вынесены за основную сетку занятий и проводятся после уроков. Элективные курсы всегда проводятся наравне с другими уроками.

Рассмотрим подробнее факультативные курсы по информатике и ИКТ.

Факультативные занятия ведутся в школе с 8 класса. Их цель - углубление и расширение знаний, развитие интереса учащихся к предмету, развитие их способностей, привитие школьникам интереса и вкуса к самостоятельным занятиям информатикой, воспитание и развитие их инициативы и творчества.

Программа факультативных занятий по информатике составляется так, что все её вопросы могут изучаться синхронно с изучением основного курса информатики в школе.

Для того чтобы факультативные занятия по информатике были эффективными, необходимо их организовать там, где есть высококвалифицированные учителя, способные вести занятия на высоком научно-методическом уровне и не менее 15 учащихся, желающих изучить факультативный курс.

При проведении факультативных занятий в основном использовать методы изучения (а не обучения) информатики, а также проблемную форму обучения.

В настоящее время факультативные занятия по информатике проводятся по трем основным направлениям:

1. Изучение программирования.
2. Изучение компьютерных технологий.
3. Изучение прикладных областей деятельности на основе компьютерных технологий.

Факультативный курс предусматривает использование следующих форм работы:

1. Фронтальной – подача учебного материала всему коллективу учеников.
2. Индивидуальной – самостоятельная работа учащихся оказанием учителем помощи при возникновении затруднения, не уменьшая активности учеников и содействуя выработки навыков самостоятельной работы.
3. Групповой – когда учащимся предоставляется возможность самостоятельно построить свою деятельность, ощутить помощь со стороны друг друга, учесть возможности каждого на конкретном этапе деятельности.

Существуют методические рекомендации [8], которые могут быть полезны для учителей, разрабатывающих новые или модифицирующих существующие учебные программы.

Структура учебной программы курса

Структура программы – форма представления курса, определяющая внутренние связи учебного материала. При выборе структуры по организации содержания учитываются цели обучения, требования к уровню обученности, характер и особенности получаемых знаний, а также особенности тех групп учащихся, которым будет адресовано разрабатываемое содержание. Наиболее значимые части содержания курса, группируются определенным образом (модули, блоки и т. п.), в соответствии с представлениями автора о характере предъявляемого материала.

Компонентами программы курса являются:

1. Титульный лист: наименование образовательного учреждения, реализующего программу; название программы; Ф.И.О. автора/составителя программы (с указанием должности и квалификационной категории); профиль и адресат программы (класс, возраст); назначение, уровень программы; сведения о статусе программы (где, когда и кем утверждена программа); сроки реализации программы; год разработки программы.).
2. Пояснительная записка (аннотация*, у*казание на место и роль курса, цель и задачи, сроки реализации программы (продолжительность обучения, этапы), основные принципы отбора и структурирования материала, методы и формы обучения, режим занятий, предполагаемые результаты).
3. Учебно-тематический план (содержание курса по темам, количество часов, в том числе аудиторной работы, консультаций, самостоятельной работы).
4. Содержание курса по темам (название темы, количество часов, выделенных на ее изучение, основные изучаемые вопросы, формы занятий, понятия, проблемы, литература).
5. Информационно-методическое сопровождение (контроль уровня обученности, перечень литературы, учебно-методическое обеспечение предмета и т. д.).

## **1.2** **Содержание линии «Алгоритмизация и программирование» в школьном курсе информатики и ИКТ**

Рассмотрим, как представлена линия «Алгоритмизация и программирование» в действующем образовательном стандарте и примерных программах.

Линия «Алгоритмизации и программирования» предназначена для изучения в 9 классе на базовом уровне. На тему «Алгоритмизация» отводится 4 часа, на тему «Программирование» – 12 часов. Данная тема входит в стандарт среднего (полного) общего образования Информатики и ИК и является частью линии алгоритмизации и программирования. В проекте стандарта и обязательном минимуме по информатике содержание алгоритмической линии определяется через следующий перечень понятий: алгоритм; свойства алгоритмов; исполнители алгоритмов; система команд исполнителя; формальное исполнение алгоритмов; основные алгоритмические конструкции; вспомогательные алгоритмы. Понятие алгоритма является центральным. Главной целью раздела алгоритмизации является овладение учащимися структурной методикой построения алгоритмов [8].

Требования к знаниям и умениям учащихся по линии алгоритмизации и программирования

Учащиеся должны знать:

* что такое алгоритм; какова роль алгоритма в системах управления;
* в чем состоят основные свойства алгоритма;
* способы записи алгоритмов: блок-схемы, учебный алгоритмический язык;
* основные алгоритмические конструкции: следование, ветвление, цикл; структуры алгоритмов;
* назначение вспомогательных алгоритмов; технологии построения сложных алгоритмов: метод последовательной детализации и сборочный (библиотечный) метод;
* основные свойства величин в алгоритмах обработки информации: что такое имя, тип, значение величины; смысл присваивания;
* назначение языков программирования;
* в чем различие между языками программирования высокого уровня и машинно-ориентированными языками;
* правила представления данных на одном из языков программирования высокого уровня (например, на Паскале);
* правила записи основных операторов: ввода, вывода, присваивания, цикла, ветвления;
* правила записи программы;
* что такое трансляция;
* назначение систем программирования;
* содержание этапов разработки программы: алгоритмизация кодирование отладка тестирование [8].

Учащиеся должны уметь:

* пользоваться языком блок-схем, понимать описания алгоритмов на учебном алгоритмическом языке;
* выполнять трассировку алгоритма для известного исполнителя;
* составлять несложные линейные, ветвящиеся и циклические алгоритмы управления одним из учебных исполнителей;
* выделять подзадачи;
* определять и использовать вспомогательные алгоритмы;
* составлять несложные программы решения вычислительных задач с целыми числами;
* программировать простой диалог;
* работать в среде одной из систем программирования (например, Турбо Паскаль);
* осуществлять отладку и тестирование программы [8].

Подходы к раскрытию темы в учебной литературе

Рассмотрим, как раскрывают содержание линии алгоритмизации и программирования авторы различных учебников информатики и ИКТ.

Рассмотрим, как представлена линия «Алгоритмизация и программирование» в школьных учебниках по информатике и ИКТ.

1. Учебник Босова Л.Л. «Информатика» 6 класс. Год издания: 2015 г. Тема «Алгоритмизация и программирование» содержится в пяти параграфах.

§14. Что такое алгоритм.

§15. Исполнители вокруг нас.

§16. Формы записи алгоритмов.

§17. Типы алгоритмов.

§18. Управление исполнителем Чертёжник.

Теоретический материал учебника поддержан развернутым аппаратом организации усвоения изучаемого материала, включающим вопросы, задачи и задания для практического выполнения, описание работ компьютерного практикума. Учебник соответствует федеральному государственному образовательному стандарту основного общего образования (2010 г.) [3].

1. Учебник Босова Л.Л. «Информатика» 8 класс. Год издания: 2015 г. Тема «Алгоритмизация и программирование» содержится в двух главах.

Глава 2. Основы алгоритмизации.

Глава 3. Начала программирования.

Теоретический материал учебника поддержан развёрнутым аппаратом организации усвоения изучаемого материала, обеспечивающим подготовку школьников к государственной итоговой аттестации по информатике в форме основного государственного экзамена (ОГЭ). Учебник соответствует федеральному государственному образовательному стандарту основного общего образования(2010 г.) [4]

1. Учебник Босова Л.Л. «Информатика» 9 класс. Год издания:   
   2015 г. Тема «Алгоритмизация и программирование» содержится в первой главе «Основы алгоритмизации и объектно-ориентированного программирования». Рассматривается в параграфах:
2. Алгоритм и его формальное исполнение.
3. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования.
4. Переменные: тип, имя, значение.
5. Арифметические, строковые и логические выражения.
6. Функции в языках объектно-ориентированного и процедурного программирования.
7. Графические возможности объектно-ориентированного языка программирования Visual Basic.

К первой главе даются практические работы компьютерного практикума.

Большое внимание в учебнике уделяется формированию у учащихся алгоритмического и системного мышления, а также практических умений и навыков в области информационных технологий. Учебник мультисистемный, так как практические работы компьютерного практикума могут выполняться в различных операционных системах: Windows или Linux. Соответствует федеральному государственному образовательному стандарту основного общего образования (2010 г.) [5].

1. Учебник Семакин И.Г. «Информатика» 9 класс. Год издания: 2015 г. Тема содержится в двух главах.

Глава I. Управление и алгоритмы.

Глава II. Введение в программирование.

В конце глав представлены дополнения.

Дополнение к главе I:

1. Автоматизированные и автоматические системы управления.
2. Использование рекурсивных процедур.

Система основных понятий главы I.

Дополнение к главе II:

1. Программирование перевода из одной системы счисления в другую.
2. Сложность алгоритмов.
3. О языках программирования и трансляторах.
4. История языков программирования.

Система основных понятий главы II.

Учебник содержит теоретический материал курса, вопросы и задания для закрепления знаний, в конце каждой главы в схематическом виде представлена система основных понятий этой главы. Некоторые главы учебника содержат дополнительный раздел, позволяющий изучить данную тему на углубленном уровне. Соответствует федеральному государственному образовательному стандарту основного общего образования (2010 г.) [11, 12].

1. Учебник Угринович Н.Д. «Информатика» 9 класс. Год издания: 2015 г. Тема «Алгоритмизация и программирование» рассматривается в первой главе «Основы алгоритмизации и объектно-ориентированного программирования», есть параграфы:
2. Алгоритм и его формальное исполнение.
3. Кодирование основных типов алгоритмических структур на языках объектно-ориентированного и процедурного программирования.
4. Переменные: тип, имя, значение.
5. Арифметические, строковые и логические выражения.
6. Функции в языках объектно-ориентированного и процедурного программирования.
7. Графические возможности объектно-ориентированного языка программирования Visual Basic.

К первой главе даются практические работы компьютерного практикума.

Большое внимание в учебнике уделяется формированию у учащихся алгоритмического и системного мышления, а также практических умений и навыков в области информационных технологий. Учебник мультисистемный, так как практические работы компьютерного практикума могут выполняться в различных операционных системах: Windows или Linux. Соответствует федеральному государственному образовательному стандарту основного общего образования (2010 г.) [1].

## **1.3** **Изучение робототехники в школе**

Согласно национальной образовательной инициативе «Наша новая школа» утвержденной Д.А. Медведевым, современное образование должно соответствовать целям опережающего развития. Для этого должно быть обеспечено:

1. Изучение не только достижений прошлого, но и технологий, которые пригодятся в будущем.
2. Обучение, ориентированное как на знаниевый, так и деятельностный аспекты содержания образования.

Таким требованиям отвечает робототехника.

«Робототехника – это область техники, связанная с разработкой и применением роботов, а также компьютерных систем для управления ими, сенсорной обратной связи и обработки информации» [6].

Введение элементов робототехники в школьные предметы позволит заинтересовать учащихся, разнообразить учебную деятельность, использовать групповые активные методы обучения, решать задачи практической направленности. Программирование реального робота поможет увидеть законы математики не на страницах тетради или учебника, а в окружающем мире. Использование конструкторов Lego Mindstorms NXT позволяет взглянуть на школьные предметы по-новому. Программирование роботов позволяет без усилий организовать межпредметные связи информатики с математикой и физикой, при специальной подготовке учителя и наличии методических материалов – с кибернетикой, физиологией и психологией.

Сегодня робототехнические конструкторы используются для проведения демонстрационных учебных экспериментов по физике, химии, биологии, математике и основам безопасности жизнедеятельности. Все это позволяет познакомить ребенка с законами реального мира и особенностями функционирования восприятия этого мира кибернетическими механизмами.

Целесообразными методами, используемыми в процессе реализации элективного курса по конструированию и программированию роботов, являются метод проектов, метод взаимообучения, модульный метод и метод проблемного обучения.

Робототехника на разных ступенях образования имеет различные цели. Поэтому рекомендуется, в зависимости от возраста учащихся, использовать конструкторы разных типов, проводить различные мероприятия, изучать всевозможные темы. Сегодня это возможно при организации специальных кружков по робототехнике, факультативов и элективных курсов.

Робототехнику, без сомнения, можно отнести к наиболее перспективным направлениям в области информационных технологий. И это не удивительно, т.к. развитие современных производств, таких, например, как автомобилестроение, микроэлектроника, станкостроение на данный момент немыслимо без использования роботизированных систем. В свою очередь, развитие подобных производств потребует подготовки большого числа специалистов в области робототехники. Что, безусловно, поставит новые задачи перед современной системой образования. Подходить к решению этого вопроса нужно комплексно. Однако решить данную задачу в рамках традиционного комплекса физико-математических дисциплин довольно сложно. Наиболее подходящей дисциплиной в этом смысле является информатика. Обучение детей робототехнике в рамках данной дисциплины может основываться наиспользовании специальных конструкторов, содержащих программируемое устройство. Наиболее распространённым на данный момент является семейство конструкторов Lego, позволяющих охватить практически все возрастные группы учащихся, начиная от младших школьников и заканчивая, учащимися старших классов.

Данное обстоятельство является крайне важным, так как позволяет сохранить преемственность и поэтапность образовательного процесса.

Условно обучение робототехнике в рамках школьного курса информатики и ИКТ может быть разделено на три этапа: начальная школа, средняя школа и старшая школа. Для обучения робототехнике в начальной школе может быть использован конструктор Lego WeDo, состоящий из стандартных деталей Lego, а также набора датчиков и приводов, подключаемых к USB. В комплекте с данным конструктором поставляется программное обеспечение, содержащее простую, интуитивно понятную среду программирования. Кроме того, вместе с набором поставляется комплект заданий, представляющих из себя 12 отдельных проектов с подробным пошаговым описанием их выполнения. Это позволяет учащемуся самостоятельно собирать и программировать действующие модели, а затем использовать их для выполнения практических задач.

Для обучения робототехнике в средней школе может быть использован конструктор Lego Mindstorms, так же состоящий из стандартных деталей Lego (планки, оси, колеса, шестерни), сенсоров, двигателей и программируемого блока NXT. Наличие отдельного программируемого блока в сочетании со средой программирования высокого уровня делает данный набор серьёзным инструментом, позволяющем создавать роботов, решающих достаточно сложные задачи. Важным достоинством Lego Mindstorms является его простота и гибкость. Набор позволяет подобрать необходимые детали практически под любую задачу либо объединить несколько наборов для решения сложных задач.

Для обучения робототехнике в старшей школе может быть использован конструктор TETRIX, являющийся основным, конструктором международных соревнований FIRST Tech Challenge. Данный конструктор состоит из набора металлических деталей, сенсоров, сервоприводов и программируемого блока NXT. Программирование роботов, собранных из данного набора, осуществляется на языке RobotC.

При введении в школьную программу курсов робототехники в учебном процессе возникают две проблемы:

1. Недостаточный уровень методических материалов.
2. Высокая стоимость одной единицы робототехнического конструктова. При этом стоит отметить, что в подавляющем большинстве случаев используются иностранные разработки.

Существует несколько сред программирования, используемых для разработки программы управления компьютером NXT [16, С. 21]:

1. NXT-G – графическая среда программирования, разработанная National Instruments для Lego, которая предоставляется в стандартном наборе конструктора. Написание программы с помощью NXT-G похоже на создание блок-схемы. Программист задает схему управления роботом, используя существующие блоки, описывающие характеристики его поведения. Данный способ создания программ является наглядным и подходит для людей с разными уровнями подготовки. Так же к преимуществам данной среды программирования можно отнести то, что она является case-редактором от Lego на базе LabView и может использовать LabView для создания «My Block». «My Block» – возможность создания собственного блока, который можно использовать в написании любой программы. Однако у NXT-G существует ряд недостатков, например, медленный редактор, недостаточная функциональность, возможность работы только с NXT.

2. Robolab – графический CASE-инструмент на базе LabView от компании National Instruments, наиболее часто используемый в школах и вузах для обучения кибернетике. В отличие от NXT-G, Robolab позволяет создавать более сложные программы с нетривиальными математическими вычислениями и использовать как числа целого типа, так и числа с плавающей точкой. Так же данный инструмент имеет довольно продвинутый механизм логирования информации. Весомыми недостатками Robolab являются отсутствие пошаговой отладки и значительная, с точки зрения российского потребителя, стоимость.

3. **TrikStudio** – коммерческая программа, ориентированная на программирование учебных роботов фирмы Trik. Включает в себя симулятор, который позволяет тестировать программы без использования реального робототехнического набора.

4. **Arduino** – среда программирования роботов на базе Arduino. Интерфейс среды разработки **Arduino** содержит следующие основные элементы: текстовый редактор для написания кода, область для вывода сообщений, текстовая консоль, панель инструментов с традиционными кнопками и главное меню. Данный софт позволяет компьютеру взаимодействовать с **Arduino** как для передачи данных, так и для прошивки кода в контроллер.

5. RobotC – текстовая среда программирования, основанная на стандартном языке C. RobotC имеет два режима: для новичков (часть функциональности языка скрыта) и для людей, имеющих опыт в программировании. Одной из возможностей среды является управление роботами с помощью джойстика. Так же данный инструмент позволяет отлаживать программу и получать обновления показаний с датчиков в режиме реального времени. Текстовое представление можно отнести как к преимуществам, поскольку позволяет полностью контролировать написание кода, так и к недостаткам, поскольку оно является не столь наглядным как графическое представление и требует умения программировать на языке С.

Cредf RobotC: предназначена как для на новичков, так и для подготовленных программистов и имеет два режима работы – базовый и расширенный. В соответствии с этим среда программирования поддерживает два языка: собственно сам RobotC, являющийся особым диалектом C (рис. 1), и «естественный язык» Natural Language, позиционирующийся как переходный от графических форм к текстовым блокам и использующий вместо низкоуровневых команд опроса датчиков и управления моторами процедуры с заранее определенными для робота действиями.

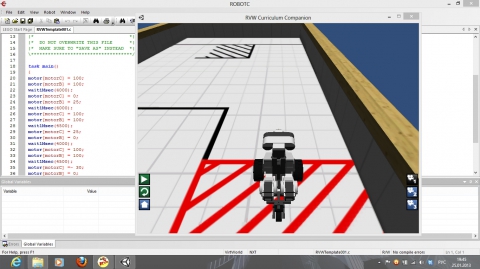


Рис. 1. Графическая среда программирования Robotc

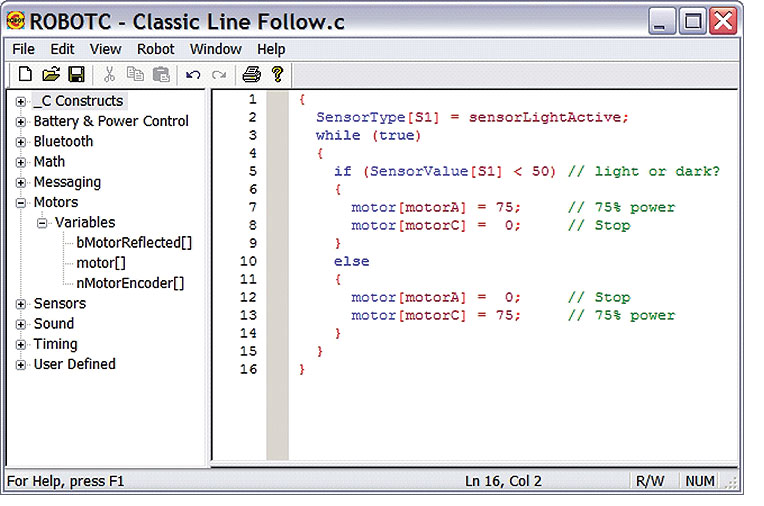


Рис. 2. Среда программирования Robotc, основанная на стандартном языке C

Кроме того, разработчики Robotc обещают в скором времени поддержку еще одного графического языка, напоминающего Scratch, в котором создание программ будет осуществляться переносом графических блоков и их соединением в пазл. Каждый блок будет иметь ряд конфигурируемых параметров, влияющих на поведение роботов.

Программное обеспечение имеет схожую с Visual Studio среду и включает в себя мощный интерактивный отладчик, способный функционировать в режиме реального времени, тем самым существенно сокращая время отладки кода. Данная среда обладает развитыми возможностями для работы с математическими выражениями, с помощью которых можно составлять весьма эффективные и сложные программы.

Среда программирования содержит все необходимые руководства, примеры и файлы справки.

# Перечислим все преимущества использования среды Robotc [15].

1. Используется индустриальный стандарт языка программирования C (Си), который дополнен специальными командами для работы с роботами.
2. Современный пользовательский интерфейс.
3. Поддерживает всесторонную отладку программ, включая пошаговое выполнение и контроль за значением переменных.
4. Дополнительные инструменты для отладки позволяют пользователям видеть в реальном времени состояния всех моторов и сенсоров. Эта информация доступна во время выполнения вашей программы, так что не нужны дополнительные усилия.
5. Более 100 примеров, которые дополняют документацию. Ученики и все заинтересованные смогут научиться программированию.
6. Расширенный редактор кода, который поддерживает одновременно несколько открытых программ, автоматическое завершение кода и многое другое.
7. В Robotc используются похожие формы и методы работы как в других профессиональных инструментах.

Команды для робота записываются в виде текста на экране, после чего компилятор Robotc обрабатывает файл и переводит текст на машинный язык. Затем файл загружается на робота, где он может быть запущен.

С педагогической точки зрения, изучение робототехники в школе имеет ряд важных достоинств. Во-первых, это стимулирование мотивации учащихся к получению знаний. При работе с Lego-конструктором учащийся видит плоды своей работы и имеет возможность применить полученные знания на практике. Кроме того, работа по созданию робота предполагает активную творческую деятельность ребёнка. Это реализуется через решение нестандартных для учащегося задач и большое количество вариантов решения.

Во-вторых, это развитие интереса учащихся к технике, программированию и конструированию. Использование подобных конструкторов в образовательном процессе ведет к популяризации профессии инженера, а также прививает учащимся интерес к робототехнике.

В-третьих, это формирование навыков программирования, развитие логического и алгоритмического мышления. В условиях информатизации образования остро встаёт необходимость поиска новых подходов к развитию алгоритмических умений школьников. Старый подход к обучению школьников программированию при помощи только языков программирования (Паскаль, Бейсик) и в редких случаях использования компьютерных исполнителей (Робот, Чертёжник и др.) уже не отвечает реалиям сегодняшнего дня. Современное образование требует более активного внедрения робототехники в курс школьной информатики.

Всё вышесказанное свидетельствует о том, что использование Lego наборов в процессе обучения информатике позволяет перейти к изучению основ робототехники, а также преодолеть недостатки традиционного подхода в обучении программированию и вывести его на новый уровень.

# **Глава 2 Обучение учащихся основам программирования с использованием среды Robotc**

## **2.1 Определение структуры и содержания курса «Робототехника. Программирование»**

Обучать учащихся основам программирования мы предлагаем в рамках факультативного курса «Робототехника. Программирование» с использованием среды Robotc.

Пояснительная записка

Программа курса составлена в соответствии с требованиями Федерального государственного образовательного стандарта начального общего образования.

Введение дополнительной образовательной программы «Робототехника. Программирование» в школе неизбежно изменит картину восприятия учащимися технических дисциплин, переводя их из разряда умозрительных в разряд прикладных. Применение учащимся на практике теоретических знаний, полученных на математике или физике, ведет к более глубокому пониманию основ, закрепляет полученные навыки, формируя образование в его наилучшем смысле. И с другой стороны, игры в роботы, в которых заблаговременно узнаются основные принципы расчетов простейших механических систем и алгоритмы их автоматического функционирования под управлением программируемых контроллеров, послужат хорошей почвой для последующего освоения сложного теоретического материала на уроках. Программирование на компьютере (например, виртуальных исполнителей) при всей его полезности для развития умственных способностей во многом уступает программированию автономного устройства, действующего в реальной окружающей среде. Подобно тому, как компьютерные игры уступают в полезности играм настоящим.

Конструктор Lego предоставляет ученикам возможность приобретать важные знания, умения и навыки в процессе создания, программирования и тестирования роботов. «Мозгом» робота Lego Mindstorms Education является микрокомпьютер Lego NXT/EV3, делающий робота программируемым, интеллектуальным, способным принимать решения. Для связи между компьютером и NXT/EV3 можно использовать также беспроводное соединение Bluetooth (плюс Wi-Fi в случае EV3). На NXT имеется три выходных порта для подключения электромоторов или ламп, помеченные буквами А, В и С. Контроллер EV3 имеет уже 4 выхода для подключения электромоторов. С помощью функции NXT/EV3Program (Программы NXT/EV3) можно осуществлять прямое программирование блока NXT/EV3 без обращения к компьютеру. Датчики получают информацию от микрокомпьютера NXT/EV3 [12].

Цель: развитие логического и алгоритмического мышления учащихся с использованием высокотехнологического учебного оборудования и ИКТ, изучения основ алгоритмизации и программирования с использованием конструктора Lego и среды программирования Robotc.

Основные задачи:

* знакомство со средой программирования RobotC;
* усвоение основ программирования, умения составления различные виды алгоритмов;
* формирование умения строить модели по схемам;
* получение практических навыков конструктивного воображения при разработке индивидуальных или совместных проектов;
* проектирование технического, программного решения идеи, и ее реализации в виде функционирующей модели;
* развитие умения ориентироваться в пространстве;
* умение использовать системы регистрации сигналов датчиков, понимание принципов обратной связи;
* проектирование роботов и программирование их действий;
* через создание собственных проектов прослеживание пользы применения роботов в реальной жизни;
* расширение области знаний о профессиях;
* умение учеников работать в группах;
* воспитание самостоятельности, аккуратности и внимательности при работе с роботом;
* планомерная подготовка к участию в робототехнических соревнованиях различного уровня (городские, областные, региональные, всероссийские).

Возраст детей, участвующих в реализации данной образовательной программы: от 13 до 16 лет (7-9 классы). Программа может быть скорректирована в зависимости от возраста учащихся. Некоторые темы взаимосвязаны со школьным курсом и могут с одной стороны служить пропедевтикой, с другой стороны опираться на него.

Практически все возможности образовательного конструктора относятся к различным компетентностным сферам, таким как к сфере самостоятельной познавательной деятельности при разработке модели робота, социально-трудовой деятельности в ходе совместной практической работе над проектом, к бытовой сфере относиться непосредственная тематика разрабатываемых проектов, к сфере культурно-досуговой деятельности могут быть причислены все из приведённых задач.

Данная классификация соответствует базовым компетентностям, которые должны быть сформированы у учащегося и развитие которых обеспечивает курс «Робототехника. Программирование».

Основными педагогическими принципами, обеспечивающими реализацию программы факультативного курса:

* + - принцип максимального разнообразия предоставленных возможностей для развития личности;
    - принцип возрастания роли внеурочной работы;
    - принцип индивидуализации и дифференциации обучения;
    - принцип свободы выбора учащимися образовательных услуг, помощи и наставничества.

В качестве платформы для создания роботов используется конструктор Lego Mindstorms NXT/EV3. На занятиях по робототехнике осуществляется работа с конструкторами серии LEGO Mindstorms. Для создания программы, по которой будет действовать модель, используется специальный язык программирования RobotC.

Курс составлен на основе опытных данных, полученных в процессе проведения факультатива «Робототехника. Программирование» на базе МБОУ СОШ №66.

Место курса «Робототехника. Программирование»

Учебный план предусматривает изучение курса «Робототехника. Программирование» в объеме 34 часов (2 часа в неделю).

Формы и методы обучения:

1. Формирование и совершенствование умений и навыков (изучение нового материала, практика).
2. Обобщение и систематизация знаний (самостоятельная работа, творческая работа, дискуссия).
3. Контроль и проверка умений и навыков (опрос).
4. Комбинированные занятия.
5. Создание ситуаций творческого поиска.
6. Стимулирование (поощрение, выставление баллов, грамоты).

Формы контроля и оценки образовательных результатов. Текущий контроль уровня усвоения материала осуществляется по результатам выполнения обучающихся практических заданий.

Итоговый контроль реализуется в форме соревнований (олимпиады) по робототехнике.

Контроль усвоения знаний курса:

– проверка проектов в среде Robotc;

– защита проектов;

– участие в соревнованиях.

Основные виды деятельности:

– знакомство с интернет-ресурсами, связанными с робототехникой;

– проектная деятельность;

– работа в парах, в группах;

– соревнования.

Формы работы, используемые на занятиях:

– лекция;

– беседа;

– демонстрация;

– практика;

– творческая работа;

– проектная деятельность.

*Ожидаемый результат*

Учащийся будет знать:

– конструкцию, органы управления и дисплей NXT/EV3;

– датчики NXT/EV3;

– интерфейс программы Robotc;

– основы программирования, программные блоки среды Robotc;

– основные виды робототехнических соревнований и особенности их проведения.

Учащийся будет уметь:

* структурировать поставленную задачу и составлять план ее решения;
* самостоятельно решать технические задачи в процессе конструирования роботов (планирование предстоящих действий, самоконтроль, применять полученные знания, приемы и опыт конструирования с использованием специальных элементов и т.д.);
* работать с литературой, с журналами, с каталогами, в интернете (изучать и обрабатывать информацию);
* составлять алгоритмы обработки информации;
* ставить задачу и видеть пути её решения;
* разрабатывать и реализовывать проект;
* проводить монтажные работы, наладку узлов и механизмов;
* собирать робота, используя различные датчики;
* программировать робота.

*Необходимые программы и материалы*

1. Конструктор Lego Mindstorms NXT/EV3.
2. Персональный компьютер/ноутбук.
3. Среда программирования RobotC.
4. Белый рабочий стол – «Игровое поле» (в качестве рабочей поверхности можно использовать белую маркерную доску).
5. Набор распечатанных на белой глянцевой бумаге, или пластике, игровых полей.

Литература для учащихся

1. Чехлова А. В., Якушкин П. А.«Конструкторы LEGO DAKTA в курсе информационных технологий. Введение в робототехнику». – М.: ИНТ, 2001 г.
2. Филиппов С.А. «Робототехника для детей и родителей». – М.: «Наука», 2010 г. – 319 с.

Литература для учителя

1. «Новые информационные технологии для образования». Институт ЮНЕСКО по информационным технологиям в образовании. Издательство «Москва». 2000 г.
2. Концепция модернизации российского образования http://www.ug.ru/02.31/t45.htm.
3. Поташник М.М. Управление профессиональным ростом учителя в современной школе. – М., 2009.
4. Руководство преподавателя по Robotc для Lego Mindstorms (30 базовых уроков).
5. Злаказов А.С., Горшков Г.А., Шевалдина С.Г. «Уроки Лего-конструирования в школе: методическое пособие». М: -БИНОМ. Лаборатория знаний, 2011. -120 с.

Интернет-ресурсы

1. http://lego.rkc-74.ru/
2. http://www.lego.com/education/
3. http://www.wroboto.org/
4. http://robosport.ru/
5. http://www.prorobot.ru/

## **2.2 Тематическое планирование факультативного курса**

В таблице 1 представлено тематическое планирование факультативного курса «Робототехника. Программирование» в средней школе, а в таблице 2 описано его содержание.

Таблица 1

Тематическое планирование факультативного курса   
«Робототехника. Программирование»

|  |  |  |
| --- | --- | --- |
| № урока | Тема и ее основное содержание | Кол-во часов |
|  | Знакомство со средой программирования RobotC | 9 |
| 1 | Среда программирования RobotC. Знакомство с  интерфейсом среды программирования. Основные особенности языка ROBOTC. | 2 |
| 2 | Структура программы на RobotC. Пример написания  простой программы, для подсчета радиуса круга.  Изучение команд ответственных за работу моторов и  движение робота. | 1 |
| 3 | Переменные в RobotC. Типы переменных. Программирование. | 1 |
| 4 | Организация цикла while. Функция, параметры, операторные скобки. Написание простых программ для движения вперед, назад, движение по квадрату, повороты. | 1 |
| 5 | Вывод на экран текстовой и графической информации | 1 |
| 6 | Организация ветвлений. Условие if. | 1 |
| 7 | Настройка RobotC на работу с сенсорами. Программирование действий роботов в зависимости от показаний сенсоров. Работа с ультразвуковым сенсором. Написание программ с использованием циклов. Вывод  на экран показаний сенсора. | 1 |
| 8 | Сборка робота машинки. Работа с сенсором оборотов.  Точное передвижение робота. | 1 |
| 9 | Задания по пройденному материалу. Самостоятельная  работа. Соревнование кегельринг. | 1 |
|  | Программирование. Работа с массивами. Графика.  Форматирование выводимых данных. | 8 |
| 10 | Типы переменных, вывод на экран различных типов переменных. Форматирование вывода. Рисование, вывод геометрических фигур. | 2 |
| 11 | Массив. Использование массивов. Задание параметров для программы с помощью массива.  Сортировка элементов массива. | 2 |
| 12 | Робот в лабиринте. Движение по лабиринту, сохранение данных о лабиринте. Вывод на экран данных о движении робота и схемы лабиринта. | 2 |
| 13 | Автоматическая 2-х скоростная коробка передач. Сборка и программирование конструкции. | 2 |
|  | Программирование. Алгоритмы сортировки. Функции | 10 |
| 14 | Сортировка данных в RobotC. Написание программы, с визуализацией работы алгоритмов. | 2 |
| 15 | Работа с массивами. Сохранение значений сенсоров в массив. Обработка массивов. | 2 |
| 16 | Параллельные задачи в RobotC. | 2 |
| 17 | Функции в RobotC. Передача параметров в функцию. Возвращаемые значения. | 2 |
| 18 | Взаимодействие роботов, передача данных через Bluetooth. | 2 |
|  | Работа над проектом. | 7 |
|  | ИТОГО | 34 |

Таблица 2

Содержание курса

|  |  |  |  |
| --- | --- | --- | --- |
| Тема | Часы | Ожидаемые знания | Навыки |
| Знакомство со средой программирования RobotC | 9 | Основные особенности языка ROBOTC. Основы программирования на языке Си, в том числе с использованием команд управления роботом. Датчики. Циклы и ветвления в RobotC.  Вывод на экран текстовой и графической информации. | Умение составить программу с выводом текстовых и графических данных на экран. Умение реализовать на текстовом и графическом языке программу. |
| Программирование. Работа с массивами. Графика  Форматирование выводимых данных | 8 | Базовые алгоритмические структуры. Способы записи алгоритмов: алгоритмический язык, псевдокод, блок-схема, программа. Основные операции языка программирования Си. Базовые типы данных. Переменные. Рекурсия. Массивы. | Решение задачи с использованием рекурсии. Использование параллельных процессов для управления роботом. Сохранение данных в массив и файл. Рисование, вывод геометрических фигур. |
| Программирование. Алгоритмы сортировки.  Функции | 10 | Сортировка данных в RobotC. Функции. Специальные команды RobotC: управление моторами, сбор данных с датчиков, подпрограммы. Элементы теории автоматического управления. Решение простейших задач слежения. Балансирующие роботы. | Уметь собрать робота без инструкции и составить программу в среде RobotC, которая обеспечит решение поставленной задачи: балансирование на двух колесах, движение по линии с препятствиями, выталкивание предметов из круга, выход из лабиринта, движение вдоль стены, объезд предметов и др. |
| Работа над проектом | 7 | Принципы организации проектной деятельности. Постановка задачи, составление плана работы, распределение ролей в команде. Принципы составления отчета и представления проекта. | Выполнить план работы, внести необходимые коррективы по ходу деятельности. Представить проект перед научным жюри. Составить отчет о выполненной работе. |
| ИТОГО | 34 |  |  |

Приведем примеры формулировок проектных заданий.

Проектное задание 1

Создать поле для Робота по образцу. Создать программу, используя датчик касания, чтобы робот попал из поля «старт» в поле «стоп» (рис. 3).

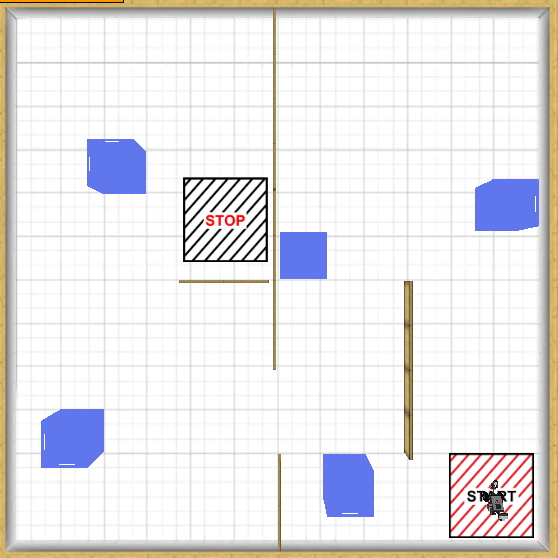


Рис. 3. Поле для робота

Проектное задание 2

Создать поле для Робота по образцу. Создать программу для Робота для прохождения восьмерки (рис. 4).

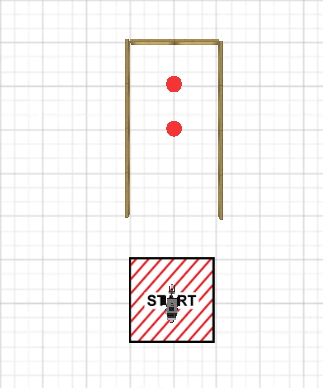


Рис. 4. Поле для робота

Проектное задание 3

Робот должен вытолкнуть все банки за пределы круга, но сам робот должен оставаться внутри (банки выталкиваются только передним датчиком), учитывается время на выполнение программы (рис. 5).

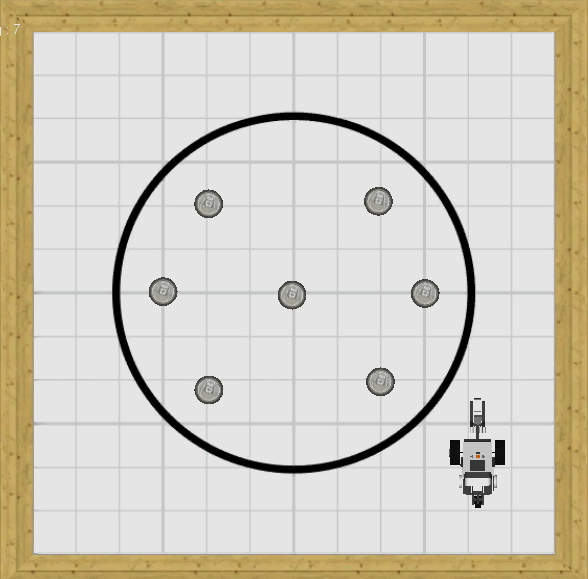


Рис. 5. Поле для робота

Проектное задание 4 (рис. 6)

1. Составить программу автоматической парковки робота в гаражный бокс 2 из позиции 1.
2. Составить программу автоматического выезда робота из гаражного бокса 2 к выходу 1.
3. Составить программу автоматического заезда автомобиля в ремонтный бокс 3 из гаражного бокса 2.
4. Составить программу автоматической парковки автомобиля:
5. Заезд в ремонтный бокс 3
6. Парковка в гаражный бокс 2.
7. Ожидание в боксе (10 секунд)
8. Выезд из гаража в позицию 1.

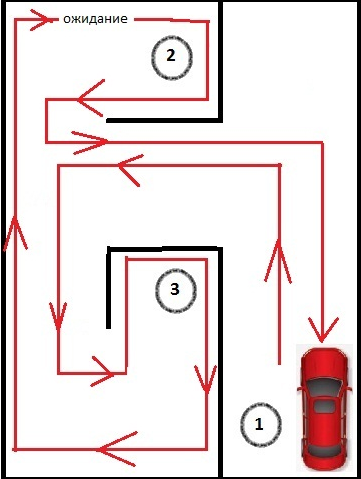


Рис. 6. Поле для робота

Проектное задание 5

Составьте программу для робота, который выполняет роль охранной сигнализации. Например, когда он обнаруживает препятствие впереди себя, он издает звук сирены.

Проектное задание 6

Создать программу движения Робота по квадрату с возвращением на базу (рис. 7).

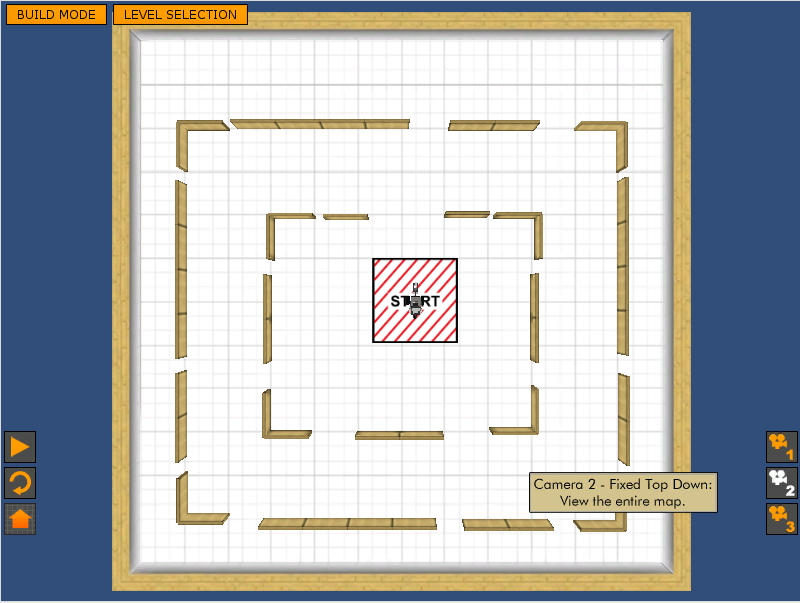


Рис. 7. Поле для робота

## **2.3 Конспекты факультативных занятий по курсу «Робототехника. Программирование»**

Конспект занятия 1

Тема: Циклы в RobotC. Работа с датчиками.

Тип: комбинированный

Учащиеся во время занятий делятся по группам по 4 человека. Каждая группа оснащена ноутбуками с установленным программным обеспечением RobotC, базовым комплектом конструктора Lego Mindstorm EV3.

Участники группы имеют свои роли: руководитель, администратор, программист и контролер времени. Роли периодически меняются, чтобы каждый участник группы «примерил» их на себе. Ребята ведут инженерные журналы определенного образца, в которых записывают весь теоретический, практический материал и ответы на вопросы учителя.

Итог занятия оценивается всеми участниками учебного процесса: учениками (взаимооценка, самооценка) и учителем. В соответствующих формах все проставляют не оценки, а баллы усвоения материала.

Окончанием занятия является анализ и общее восприятие командой и в целом класса данного на уроке материала.

Цель:

1. Применять такую синтаксическую конструкцию языка Robotc, как [цикл while](http://srv.innopolis.edu.ru:8080/mod/resource/view.php?id=769).
2. Писать простейшие логические конструкции на языке Robotc и понимать их смысл.

Задачи:

1. Кратко формулировать, почему датчики являются важной составляющей робототехнических систем.
2. Использовать для управления роботом модель «восприятие-планирование-действие».
3. Настраивать и использовать датчик касания робота.
4. Сравнивать и сопоставлять программы с циклом while с программами, не имеющими этой конструкции, управляющей потоком программы.

Средства обучения:

* интерактивная доска;
* ноутбуки (ПК), с установленным программным обеспечением Robotс;
* базовый комплект конструктора Lego Mindstorm EV3;
* инструкция по сборке EV3.

Ход занятия

Оргмомент

Приветствие. Подготовка к занятию.

Актуализация знаний

Учитель: На предыдущих уроках мы с вами говорили о роботах, которые отличаются по внешнему виду, их функциональным возможностям и т.д.

Вспомним роботов, которых мы конструировали с вами на предыдущих занятиях. Как двигаются роботы, собранные нами на предыдущих занятиях?

Ученики: Роботы двигаются по времени, на заданное расстояние; по заданной траектории, переносят предметы с определенного месторасположения.

Учитель: Предлагаю просмотреть изображения и сделать анализ, ответив на вопросы (рис. 8).

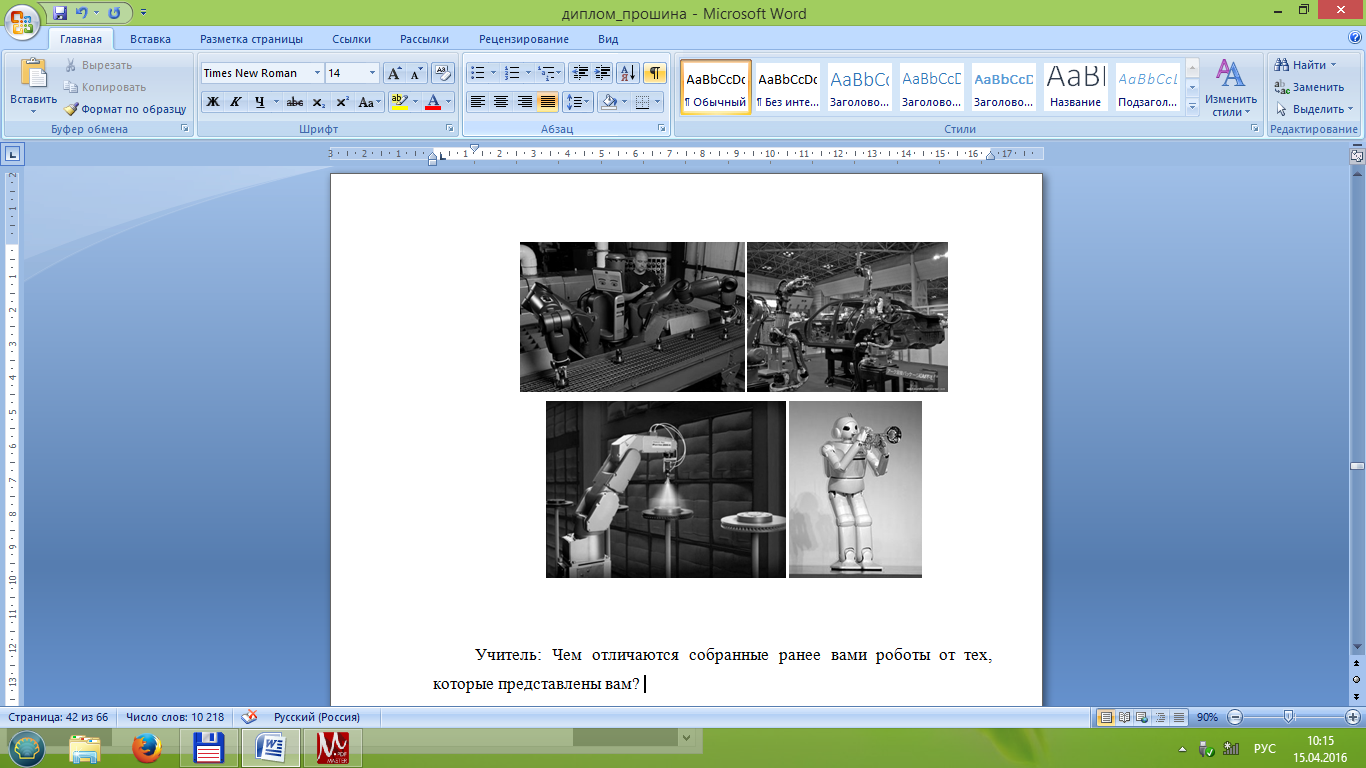


Рис. 8. Роботы

Учитель: Чем отличаются собранные ранее вами роботы от тех, которые представлены вам?

Ученики: Роботы, которые раньше собирали, работали только в зависимости от времени или четко выверенных формул углов поворота, движений вперед и механических захватов. У роботов с фотографий есть датчики: касания, расстояния, цвета/света/освещенности.

Учитель: Почему нельзя обойтись без датчиков?

Ученики: Потому что без датчиков будет долгая и трудоемкая работа, мы не можем знать размеров исследуемого объекта, расстояния до него. Датчики позволяют определять дополнительные параметры исследуемого объекта.

Объяснение нового материала

Учитель: Итак, определим ключевые датчики, используемые при работе роботов?

Ученики: Датчики касания, определения расстояния, цвета/света/освещенности, температуры и т.д.

Учитель: Перед вами датчики, которые входят в базовый комплект EV3 (рис. 9 а), б), в)).

|  |  |  |
| --- | --- | --- |
| Развивающие кубики LEGO Education для детей, игры с кубиками Петербург набор Образование Лего 45507 СПб 45507 Датчик касания EV3  датчик касания; | Лего MINDSTORMS EV3 (Lego MINDSTORMS EV3)  датчик   расстояния; | 45506 DoubleBrick: Российское сообщество энтузиастов LEGO!  датчик цвета/света/  освещенности. |
| а) | б) | в) |

Рис. 9. Датчики роботов Lego

Учитель: Сегодня мы познакомимся с датчиком касания. Прикрепите его к своим тележкам (ученики прикрепляют датчики касания к своим тележкам).

Для чего нужен данный датчик? Каков принцип работы датчика касания?

Ученики: В зависимости от нажатия кнопки касания на датчике, робот может двигаться вперед или назад, останавливаться или выполнять другие манипуляции. Все зависит от условий, предъявляемых к датчику.

Учитель: Иначе говоря, робот работает, ПОКА нажата или отжата кнопка касания. Делая аналогию с языком программирования Pascal, работа датчика касания зависит от организации цикла ПОКА (WHILE).

Организация цикла ПОКА очень похожа на известный цикл в языком программирования Pascal (табл. 3).

Таблица 3

Организация цикла ПОКА

|  |  |
| --- | --- |
| Язык программирования Pascal | Язык программирования RobotC |
| While (<условие>) do  begin  <Тело цикла>  end; | While (<условие>)  {  <Тело цикла>  } |

Учитель: Выясним, что должно быть в <условии> и внутри <тела цикла>.

Ученики: В условии должно отображаться состояние датчика касания- нажата или отжата кнопка.

Учитель: Совершенно верно, для того, чтобы идентифицировать датчик касания, необходимо войти во вкладку Motors and Sensors Setup и просмотреть индекс датчика (рис. 10).

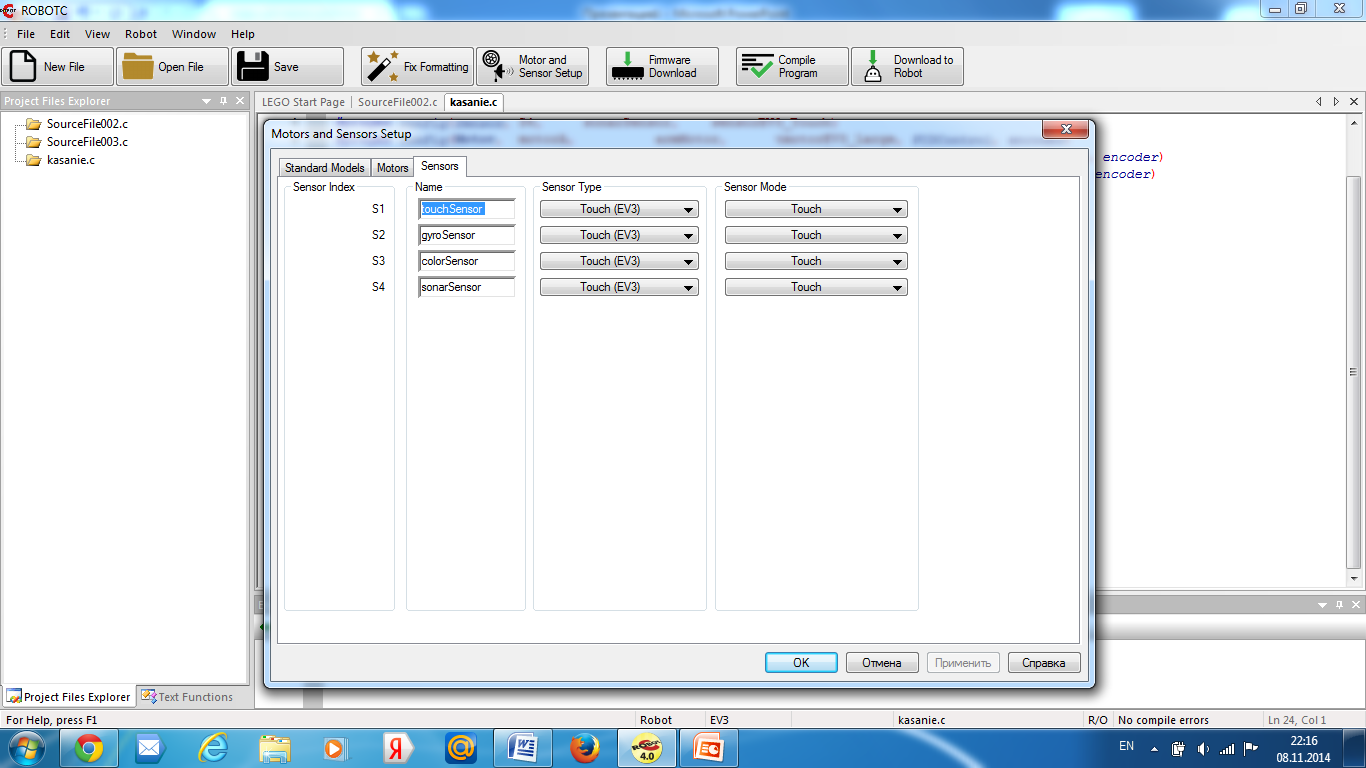


Рис. 10. Идентификация датчика касания

Условие будет выглядеть так: SensorValue(touchSensor) = = 0 или SensorValue(S1) = = 0

Учитель: Давайте подробно разберем организацию цикла While (табл. 4). Что происходит в результате выполнения программы к примерам 1, 2. (Каждый пример рассматривается отдельно с подробным объяснением учителя. После разбора каждого примера, учащиеся пробуют программы на своих тележках).

Таблица 4

Пример программ с разными циклами

|  |  |
| --- | --- |
| Цикл с условием выхода | Бесконечный цикл  Условия, при которых цикл станет бесконечным: 1= =1 или true |
| task main()  {  while (SensorValue(touchSensor) = = 0)  {  motor[motorB]=60;  motor[motorC]=60;  }  motor[motorB]=0;  motor[motorC]=0;  } | task main()  {  while (true)  {  while (SensorValue(touchSensor) = = 1)  {  motor[motorB]=-60;  motor[motorC]=-60;  }  }  motor[motorB]=0;  motor[motorC]=0;  } |
| Робот движется, пока датчик касания не нажат. Наталкиваясь на препятствие, робот останавливается. | Робот движется при нажатой кнопке касания и останавливается, если кнопка отжата. (Имитация пульта управления) |

Закрепление материала

Пример: Используя два датчика касания, создайте робота, который будет поворачивать налево или направо в зависимости от нажатой кнопки датчика.

Примерная программа учеников:

task main()

{

while (true)

{

while (SensorValue(S4)==1)

{ motor[motorB]=0;

motor[motorC]=-60;}

motor[motorB]=0;

motor[motorC]=0;

while (SensorValue(S1)==1)

{ motor[motorB]=-60;

motor[motorC]=0;}

motor[motorB]=0;

motor[motorC]=0;

}

}

Задание. Модифицируйте программу «Движение вдоль черной линии» таким образом, чтобы робот двигался по ней:

а) пока он не совершит 1,5 оборота колеса;

б) пока он «не услышит» громкий звук.

Задание. Составить программу для робота, который издает звуковой сигнал при нахождении зеленого кубика

Рефлексия. Итог занятия

Итоги занятия учащиеся подводят самостоятельно, заполняя инженерные журналы и опросные листы по выполненному заданию, по совместной работе.

Конспект занятия 2

Тема: Организация ветвлений в Robotc.

Тип занятия: комбинированный.

Цель урока: формирование навыка решения задач с помощью операторов ветвления.

Задачи:

1. Образовательная: рассмотреть понятие алгоритма ветвления, виды его конструкций и соответствующие им команды на языке программирования RobotC; закрепить умения строить собственные модели роботов в зависимости от поставленной цели.
2. Развивающая: развитие творческих способностей, логического мышления, моторики учащихся.
3. Воспитательная: формирование информационной культуры учащихся, культуры труда, дисциплинированности, усидчивости, положительной мотивации учебного труда, воспитание целеустремленности;

Оборудование: доска, проектор, компьютер.

Тип урока: комбинированный

Формы организации деятельности: индивидуальная, фронтальная.

Методы: беседа, фронтальный опрос.

Оборудование: лего-конструкторы MindstormsNXT 2.0 и EV 3, ПК с установленной средой программирования Robotc, проектор, поля.

Этапы занятия:

1. Организационный момент.
2. Повторение и закрепление ранее полученных знаний.
3. Изучение нового материала
4. Практическая работа
5. Подведение итогов занятия.

Ход занятия:

Организационный момент.

Учитель: Добрый день, ребята! На прошлом уроке мы занимались конструированием роботов из конструктора NXT и EV3. Сегодня мы будем создавать для них программы, которые он будут выполнять движение вдоль четной линии.

Повторение теоретического материала.

Учитель: Для начала давайте вспомним, какие виды алгоритмов мы знаем?

Ученики: линейный, циклический и разветвляющийся.

Учитель: Для какого вида алгоритмов используются следующие блоки из среды программирования NXT-G и EV3?

Ученики: движение, цикл, ветвление.

Учитель: Какие действия может выполнить робот с помощью блока «Движение»?

Ученики: ехать вперед, ехать назад, поворот влево, поворот вправо, свободный поворот (любой угол), разворот.

Учитель: Для чего предназначен блок «Цикл» и «переключатель»?

Учитель: Что мы изучали на прошлом занятии?

Ученики: движение по черной линии с помощью одного датчика освещенности.

Учитель: Как вы думаете, всегда ли достаточно одного датчика освещенности (цвета) для движения по траектории.

Ученики: нет, не всегда.

Учитель: Как можно выйти из этой ситуации?

Ученики: поставить еще один датчик освещенности.

Изучение нового материала

Учитель: Сегодня мы изучим движение робота по черной линии с помощью датчика освещенности LightSensor.

Может ли робот двигаться по линии, то есть так, чтобы сенсор всегда находился на черном поле? Для этого представим себе, что через некоторое время после старта сенсор «уехал» с черного поля и оказался на белом. Мы понимаем, что в этом случае робот должен немного подрулить, чтобы вернуться обратно на черную линию. Но куда именно подрулить: направо или налево? Дело в том, что белый цвет по обе стороны от черной линии одинаков. Поэтому, попав в такую ситуацию, робот должен двигаться не по линии, а по границе черного и белого. Что при этом изменится? Робот и человек по-разному воспринимают границу линии. На рис. 11 мы видим, как четную линию видит человек, а как работ.

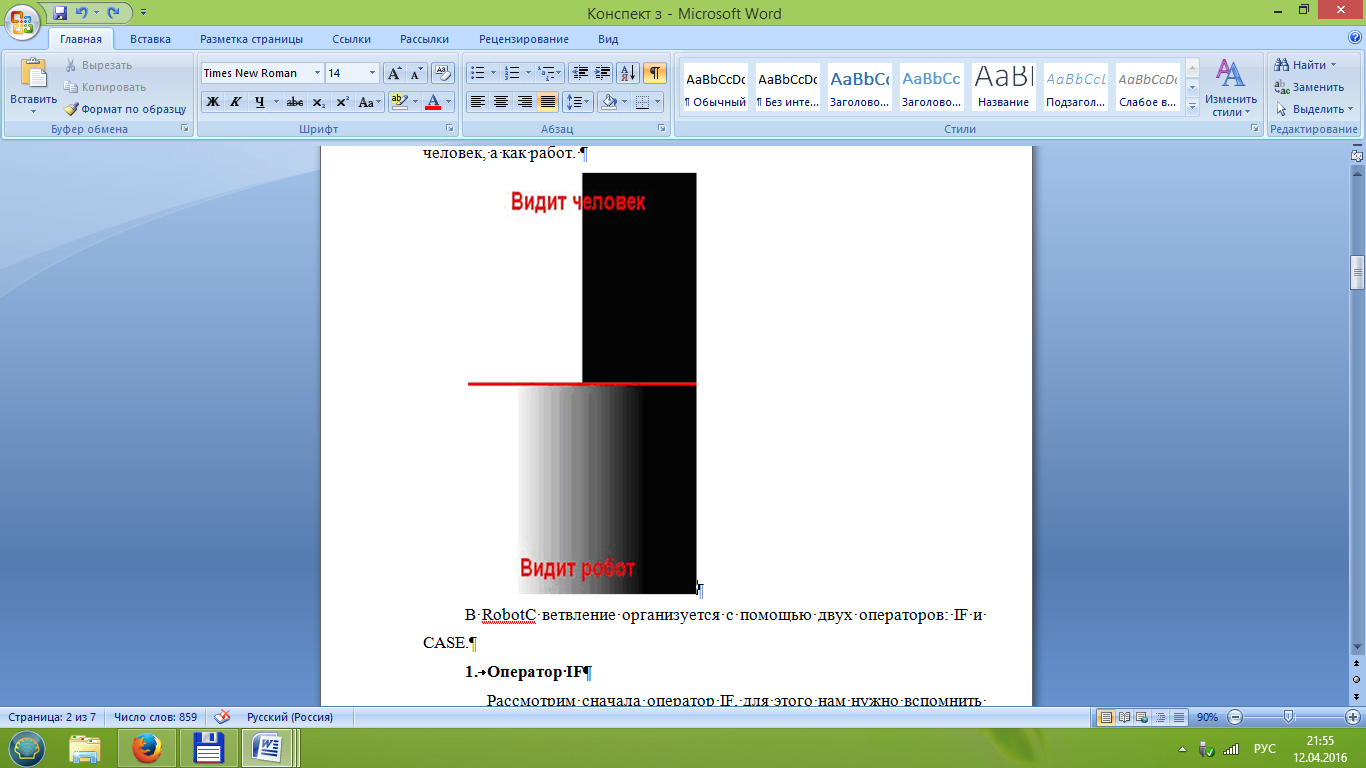
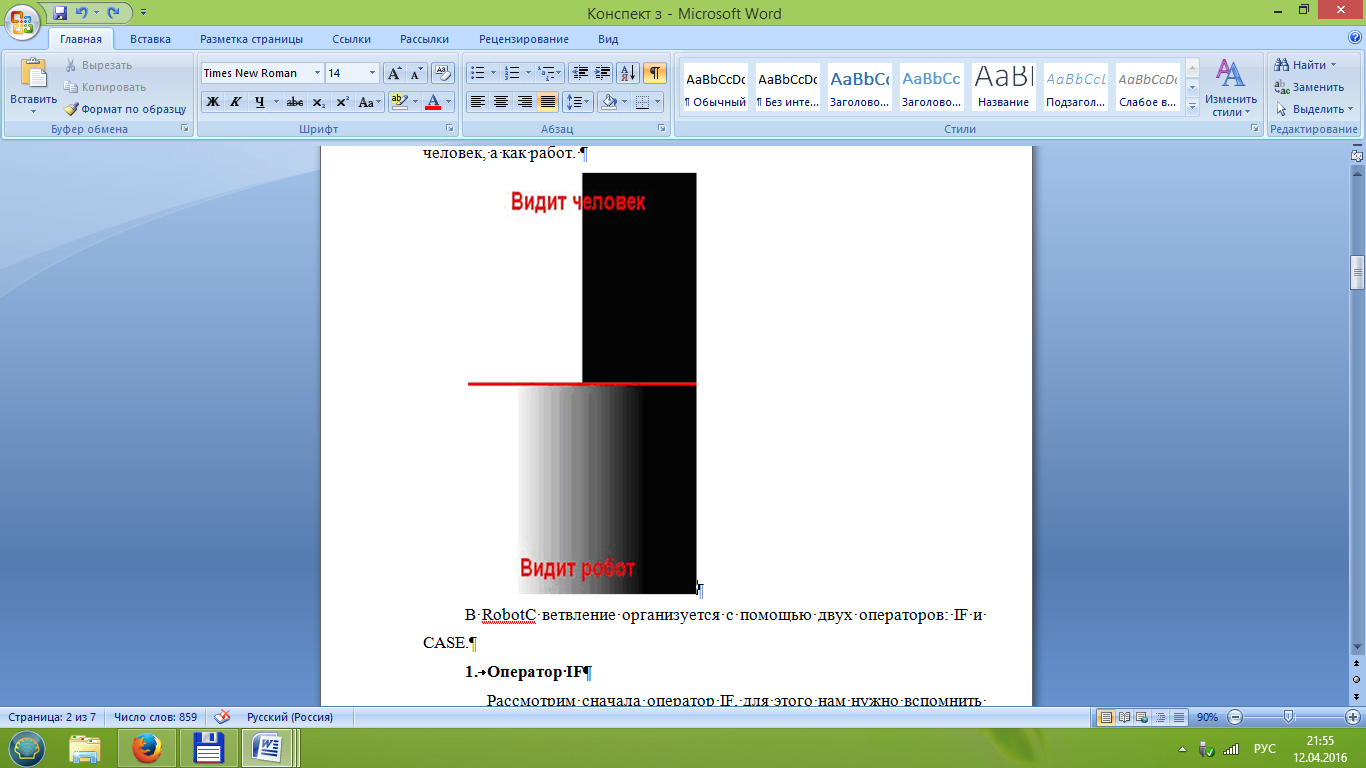


Рис. 11. Черная линия

Мы видим, что датчик света при своем движении влево или вправо не наблюдает резкой границы черного и белого. Это происходит из-за того, что при его перемещении, например, слева направо, количество света, попадающего в сенсор, постепенно увеличивается. Значит, показания датчика также будут плавно возрастать.

Робот постоянно «подруливает» то влево, то вправо в зигзагообразном стиле, когда нарушается условие пороговое значение освещенности.

значение освещенности на белом поле = 75%

значение освещенности на черном поле = 15%

Сложить полученные значения: получить среднее делением пополам (90/2=45).

Пороговое значение для датчика освещенности = 45%

SensorValue(lightSensor) < 45

В RobotC ветвление организуется с помощью оператора IF.

Рассмотрим сначала оператор IF, для этого нам нужно вспомнить какие виды (формы) ветвления бывают?

Ученики: полное и неполное ветвление (табл. 5).

Таблица 5

Виды ветвлений

|  |  |
| --- | --- |
| Не полная форма | Полная форма |
| Если…то… | Если… то…иначе… |
| условие  действие1 | условие  действие1  действие2 |

В среде RobotC условный оператор записывается следующим образом:

If (условие)

{

// команды когда условие истинны}

Else

{

// команды когда условие ложны}

При программировании модели робота следует учесть следующие моменты:

1. Если оба датчика над белой поверхностью, то это нормальная ситуация, в которой линия находится между датчиками, поэтому робот должен ехать прямо.
2. Если левый датчик еще над светлой поверхностью, а правый датчик уже над темной, значит, робот заехал своей правой частью на линию и ему нужно поворачивать направо, чтобы линия опять оказалась между датчиками.
3. Если левый датчик оказался над темной поверхностью, а правый еще над светлой, то для выравнивания роботу нужно поворачивать налево.
4. Если оба датчика над темной поверхностью, то в общем случае, робот опять продолжает двигаться прямо.

Рассмотрим программу для движения робота по траектории с двумя датчиками освещенности (цвета).

Данная конструкция определяет, как включить моторы в зависимости от показаний датчиков в определенном месте поля, соответственно через мгновение программа должна проверить, не изменились ли показания, чтобы соответствующим образом подправить поведение моторов, а через мгновение еще раз, еще и т.д. Поэтому она должна быть помещена в цикл, который будет обеспечивать такую повторяющуюся проверку. Выбираем, какой датчик будет опрашиваться первым. Это не имеет большого значения, поэтому пусть будет левый. Необходимо определить, над светлой или над темной он поверхностью.

Это действие еще не позволяет сказать в какую сторону роботу надо ехать. Но оно разделит состояния, перечисленные выше, на две группы: (I, II) для верхней ветви и (III, IV) для нижней.

В каждой из групп теперь по два состояния, поэтому необходимо выбрать какое-то из них. Если внимательно посмотреть на первые два состояния I и II, то они отличаются положением правого датчика - в одном случае он над светлой поверхностью, в другом - над темной. Именно это и определит выбор, какое действие предпринять.

Теперь нужно вставить блоки, определяющие поведение моторов: верхняя ветвь вложенного условия определяет комбинацию «оба датчика на светлом», верхняя – «левый на светлом, правый на темном».

Получившиеся две ветви наполняются блоками движения. Верхняя ветвь отвечает за состояние «левый на темном, правый на светлом», а нижняя – за «оба датчика на темном».

Нижняя ветка основного условия отвечает за другую группу состояний III и IV. Эти два состояния также отличаются друг от друга уровнем освещенности, который улавливает правый датчик. Значит, он будет определять выбор каждого из них.

Нас интересует самое нижнее положение, когда оба датчика на темном. Нам потребуется переменная логического типа.

Теперь давайте обратимся к нашим роботам (на данном уроке это колесные роботы), которые мы собирали на прошлых занятиях. Создание программы для робота.

task main()

{

wait1Msec(2000);

while(true)

{

if (SensorValue(lightSensor) < 45)

{

motor[motorB] = 60;

motor[motorC] = 20;

}

else

{

motor[motorB] = 20;

motor[motorC] = 60;

}

}

}

Практическая работа.

Создать программу«Простое движение вдоль линии». Перед стартом робот ставится на поле так, чтобы датчик освещенности «смотрел» на границу линии: черное – слева, белое – справа.

Сначала определим, какой датчик нам будет нужен для движения по черной линии? (Освещенности) Так как наборов у нас два вида, поэтому мы будем работать в двух командах. Сейчас вам нужно будет установить датчики освещенности спереди робота, между двумя колесами. Как вы думаете, какие блоки вам нужны будут для движения по черной линии? (Движение, цикл, переключатель).

Нам нужно будет узнать процент освещенности на белом участке и на черном. Для этого включите ваших роботов и узнайте это. Найдите сумму этих показателей и разделите на два, это и будет процент освещенности, который вам нужно будет установить в программе. Сейчас присаживайтесь за компьютеры, напишите программу для движения робота по черной линии (рис. 12).

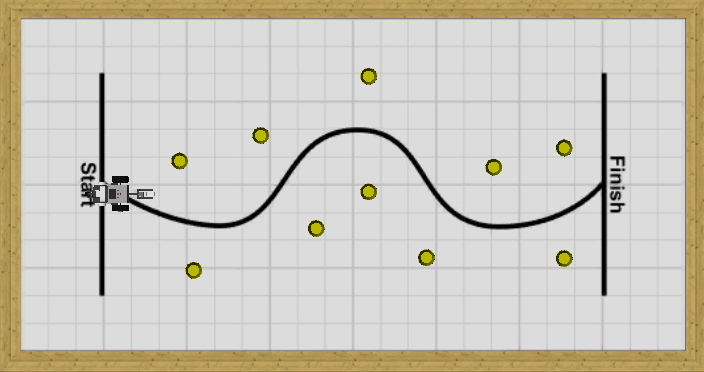


Рис. 12. Поле «Черная линия»

Презентация роботов

Каждая команда по очереди включает программу своего робота и проверяет его работоспособность.

Подведение итогов урока. Рефлексия.

Итак, ребята, давайте подведем итоги нашей работы.

Какие виды алгоритмов мы с вами сегодня использовали при составлении программ?

Какие блоки необходимы для движения по черной линии

Что в работе вызвало у вас затруднения?

Где в дальнейшем могут быть использованы знания, полученные вами сегодня на занятии?

# **Заключение**

Для того чтобы сегодня у ученика формировалась учебная успешность, нужно добиться, прежде всего, чтобы ученик осознавал, что учебная деятельность, которой он занят в данный момент в школе повлечет за собой успех в его дальнейшей деятельности. Есть много образовательных технологий развивающих критическое мышление и умение решать задачи, однако существует очень мало привлекательных образовательных сред, вдохновляющих следующее поколение к новаторству через науку, технологию, математику, поощряющих детей думать творчески, анализировать ситуацию, критически мыслить, применять свои навыки для решения проблем реального мира.

Робототехника в школе представляет учащимся технологии 21 века, способствует развитию их коммуникативных способностей, развивает навыки взаимодействия, самостоятельности при принятии решений, раскрывает их творческий потенциал. Ученики лучше понимают, когда они что-либо самостоятельно создают или изобретают. При проведении занятий и мероприятий по робототехнике этот факт не просто учитывается, а реально используется.

В данной работе проведен анализ современного состояния курсов программирования и робототехники в средней школе. Сделан вывод, что на данный момент эти два направления изучаются независимо друг от друга. Как правило, курс робототехники рассматривается на факультативных занятиях в рамках предмета «Технология». Поэтому основной упор в нем сделан на процесс конструирования. Но робототехника – это не только создание конструкции робота. Конструирование должно идти в неразрывной связи с программированием. Поэтому мы попытались соединить робототехнику и программирование в рамках одного факультативного курса.Для этого используем среду программирования RoborC.

Результатом работы является программа факультативного курса «Робототехника. Программирование» для школы (7-9 классы). Курс универсальный, рассчитан на 34 часа. В результате изучения, учащиеся будут уметь не только конструировать роботов, но и составлять алгоритмы различной сложности для управления роботами и выполнения ими различных действий. Кроме того, улучшится понимание в целом основных алгоритмических структур и повысится интерес к изучению линии «Алгоритмизация и программирование» в основном курсе информатики и ИКТ.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Федеральный государственный образовательный стандарт основного общего образования. – URL: http://минобрнауки.рф/документы/938. (дата обращения: 15.01.16).
2. Федеральный центр информационно-образовательных ресурсов [Электронный ресурс]. – URL: [http://fcior.edu.ru](http://fcior.edu.ru/) (дата обращения: 11.12.15).
3. Босова Л. Л. Информатика: учебник для 6 класса / Босова Л.Л., Босова А.Ю. – М.: БИНОМ. Лаборатория знаний, 2015. – 216 с.
4. Босова Л. Л. Информатика: учебник для 8 класса / Босова Л.Л., Босова А.Ю. – М.: БИНОМ. Лаборатория знаний, 2015. – 186 с.
5. Босова Л. Л. Информатика: учебник для 9 класса / Босова Л.Л., Босова А.Ю. – М.: БИНОМ. Лаборатория знаний, 2015. – 184 с.
6. Ершов М. Г. Роль образовательной робототехники в формировании инженерного мышления школьников [Электронный ресурс]. – URL: <http://confer.cschool.perm.ru/tezis/Ershov.doc> (дата обращения: 17.01.16).
7. Захарова И. Г. Информационные технологии в образовании: учебное пособие. – М., 2003. – 183 с.
8. Лапчик М. П. Методика преподавания информатики: учеб. Пособие для студ. Пед. Вузов /М. П. Лапчик, И. Р. Семакин, Е. К. Хеннер; под общей редакцией М. П. Лапчика. – М.: Издательский центр Академия, 2011. – 562 с.
9. Малев В.В. Общая методика преподавания информатики. – URL: <http://window.edu.ru/resource/874/37874/files/mpi_6.pdf#1>. (дата обращения: 5.02.16).
10. Сумин Д.В. Подготовка учащихся 5-7 классов к олимпиаде по робототехнике. – URL: http://www.scienceforum.ru/2014/462/4035. (дата обращения: 1.04.16)
11. Семакин И. Г. Информатика: учебник для 8 класса / И.Г. Семакин, Л.А. Залогова, С.В. Русаков, Л.В. Шестакова. – М.: БИНОМ. Лаборатория знаний, 2012. – 176 с.
12. Семакин И. Г., Шеина Т. Ю. Преподавание базового курса информатики в средней школе: метод. пособие. – М.: БИНОМ. Лаборатория знаний, 2005. – 245 с.
13. Угринович Н. Д. Информатика: учебник для 9 класса. – М.: БИНОМ. Лаборатория знаний, 2014. –152 с.
14. Ушаков А.А. Робототехника в средней школе – практика и перспективы. – URL: [www.uni-altai.ru/info/journal/vesnik/3365-nomer-1-2010.html](http://www.uni-altai.ru/info/journal/vesnik/3365-nomer-1-2010.html) (дата обращения: 15.04.15).
15. Филиппов С.А. Робототехника для детей и родителей. – М.: Наука, 2010г. – 319 с.
16. Шимов В.И. Применение робототехнических устройств в обучении программированию [Электронный ресурс]. – URL: <http://cyberleninka.ru/article/n/primenenie-robototehnicheskih-ustroystv-v-obuchenii-programmirovaniyu-shkolnikov> (дата обращения: 16.03.16).

# **Приложение**

Задачи для факультатива робототехники

Линейный алгоритм

Задача 1.

Робот находится в центре окружности диаметром не менее 40 см. С помощью коротких отрезков окружность разделена на восемь равных частей (рис. 1).

Рис. 1. Поле для робота

Задание:

Ответить на вопрос: на сколько градусов должен провернуться вал левого двигателя, чтобы робот повернулся вправо на угол в:

а) 45 градусов б) 90 градусов в) 180 градусов?

Провести экспериментальную проверку, написав программы поворота робота на указанные углы.

Запустите программы несколько раз, какова погрешность движения робота? Насколько отличаются углы поворота робота при выполнении одной и той же программы? Заполните таблицу 1.

Таблица 1.

Задание

|  |  |
| --- | --- |
| Угол поворота корпуса робота | Угол поворота левого колеса робота |
| 450 |  |
| 900 |  |
| 1800 |  |
| 10 |  |

Окружность используется в качестве транспортира.

Задача 2

Робот находится в начале отрезка черной линии длиной не менее 60 см. На расстоянии 10, 25, 40 и 60 см от начала отрезка расположены жирные, хорошо заметные черные точки.

Задание:

На сколько градусов должен повернуться вал левого и правого двигателя, чтобы робот проехал вперед на:

а) 10 см

б) 25 см

в) 40 см

г) 60 см?

Провести экспериментальную проверку, написав программы движения работа на указанные расстояния.

На жирные точки, расположенные на черной линии, устанавливаются флажки, сделанные из деталей лего-конструктора. Программы считаются правильными, если робот, начав движение от начала линии, останавливается не далее 2 см от соответствующего флажка. Заполните таблицу 2.

Таблица 2.

Задание

|  |  |
| --- | --- |
| Расстояние пройденное роботов | Угол поворота левого и правого колеса робота |
| 10 см |  |
| 25 см |  |
| 40 см |  |
| 60 см |  |

Задача 3

Робот находится в центре пересечения двух линий по 60 см длины каждая. На конце каждой линии стоит флажок, сделанный из деталей лего-конструктора (рис. 2).

Рис. 2. Поле для робота

Задание:

Написать программу движения робота вдоль линий таким образом, чтобы робот коснулся каждого флажка, не опрокинув его.

Ограничения

Робот не должен выезжать за пределы траектории обозначенной линиями.

Задача должна быть решена без использования датчиков расстояния и освещенности.

Задача 4

На игровом поле в вершинах воображаемого квадрата со стороной 60 см стоят флажки, сделанные из деталей лего-конструктора.

Робот устанавливается автором программы самостоятельно, таким образом, чтобы он находился рядом с одним из флажков с внутренней стороны квадрата (рис. 3).

Рис. 3. Поле для робота

Задание:

Написать программу движения робота вдоль периметра квадрата, таким образом, чтобы он обогнул все четыре флажка, не задев их, но и не удаляясь от стороны квадрата более чем на 20 см. Задание считается выполненным, если робот вернулся в начальную точку движения с погрешностью не более 5-10см. Пересечение воображаемой линии соединяющей вершины квадрата считается недопустимым.

Задача 5

Игровое поле свободно от посторонних предметов.

Задание:

Написать программу движения робота по дугообразной траектории (рис. 4).

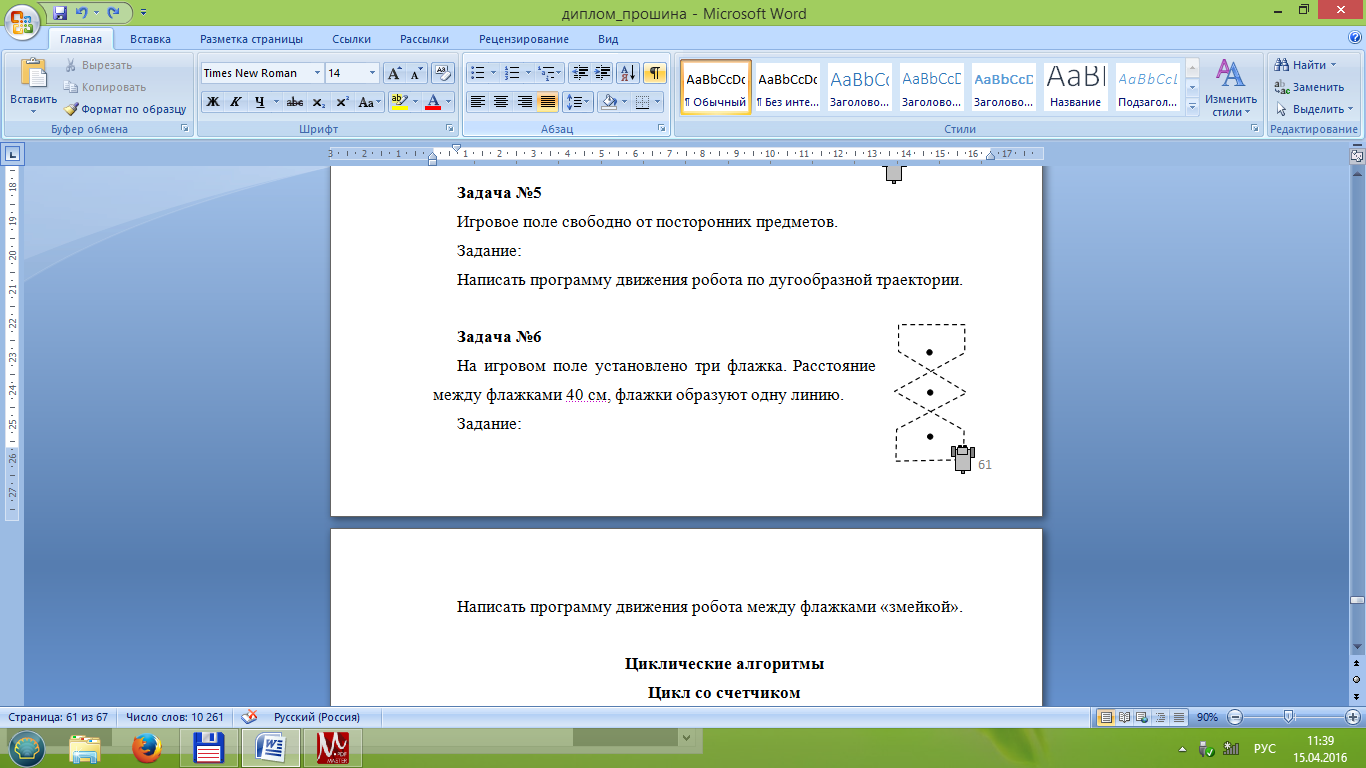


Рис. 4. Поле для робота

Задача 6

На игровом поле установлено три флажка. Расстояние между флажками 40 см, флажки образуют одну линию.

Задание:

Написать программу движения робота между флажками «змейкой».

# 

# Циклические алгоритмы

## Цикл со счетчиком

Задача 1

Используя команду цикла, усовершенствуйте решение задачи 1.3. Поставить в теле цикла звуковую команду, воспроизводящую слово «Yes», после каждого касания роботом флажка.

Задача 2

Используя команду цикла, усовершенствуйте решение задачи 1.4. Поставить в теле цикла звуковую команду, воспроизводящую слово «Yes», после каждого поворота, который производит робот. При движении по прямой робот должен воспроизводить на своем дисплее изображении «Bomb», при повороте – изображение «Boom».

Задача 3

Используя команду цикла, усовершенствуйте решение задачи 1.6. Найдите повторяющийся элемент траектории.

Задача 4

Напишите программу, которая воспроизводит следующий алгоритм:

* 1. Робот движется вперед на 10 см.
  2. Раскрывает клешни.
  3. Воспроизводит звуковой сигнал.
  4. Закрывает клешни.
  5. Пятится назад в первоначальную точку.
  6. Поворачивает вправо.
  7. Повторяет все действия 8 раз.

На какой угол должен поворачивать робот вправо, чтобы в конце выполнения программы вернуться в первоначальное положение?

## 

## Цикл с контролируемый сигналом от сенсоров

Задача 1

Робот находится на игровом поле. На расстоянии 100 см от него в зоне видимости его радаров находится небольшая картонная коробка.

Задание:

Написать программу движения робота вперед до тех пор, пока расстояние до коробки не уменьшится до 20 см. Совершать повороты роботу не потребуется.

Задача 2

Робот находится на игровом поле. На расстоянии 80см от него находится небольшая картонная коробка. Угол поворота робота относительно коробки произвольный.

Задание:

Написать программу, которая будет поворачивать робота вправо до тех пор, пока в поле зрения его радаров не окажется коробка. После остановки робота линия его взгляда должна как можно точнее пересекаться с коробкой. «Найдя» коробку робот должен сказать «Yes».

Задача 3

На белом игровом поле нарисован черный круг диаметром 60 см. Робот находится в центре круга.

Задание:

Написать программу движения робота внутри черного круга. Робот должен ехать вперед, пока под ним черный цвет и поворачивать вправо, если нет черного цвета. Движение робота должно продолжаться 60 секунд. Программа должна использовать не менее двух циклов.

# Ветвление

Задача 1

Робот должен разжимать клешни, если к его радару на расстояние 10 см поднести руку и сжимать, если рука исчезает из поля его зрения. Программа должна работать ровно 60 секунд.

Задача 2

На расстоянии 60 см от робота находится подставка с мячиком красного цвета. Подставка собрана согласно инструкции прилагаемой к конструктору Lego Mindstorms. Мячик находится в поле зрения робота, однако, угол поворота робота установлен неточно. Других предметов на игровом поле нет.

Задание:

Робот должен подъехать к мячику, взять его клешнями, развернуться и вернувшись на первоначальное место, разжать клешни.

Ожидается, что в процессе движения к мячику роботу придется несколько раз скорректировать свой маршрут.

Задача 3

На рабочем столе лежит карта из белой бумаги, на которой нарисована толстая черная линия произвольной формы. Толщина линии не менее 2-3 см. Линия не имеет пересечений. Повороты образуют угол не менее 1200. Радиус поворота линии не менее 20 см.

Задание:

Написать программу движения робота по черной линии. Робот должен двигаться, отслеживая все ее повороты.

Задача 4

На рабочем столе лежит карта из белой бумаги, на которой нарисована черная окружность диаметром 100 см. Толщина линии 3 см. Внутренняя часть круга белого цвета. На расстоянии 5 см от линии, внутри круга, на равном расстоянии друг от друга стоят пластиковые стаканчики объемом 0,33.

Задание:

Робот должен вытолкнуть все стаканчики за пределы круга за наименьшее время. Способы выталкивания произвольные.

Задача основана на классической задаче с робототехнических соревнований «Кегельринг».

# Математические операции

Задача 1

Робот стоит на игровом столе. Перед роботом строго по оси его взгляда, но на неизвестном расстоянии, находится картонная коробка. Расстояние до коробки от 20 до 150 см. За роботом на расстоянии 5 см стоит флажок.

Задание:

Робот должен измерить расстояние до коробки и сохранить показания в переменной. Одним непрерывным движением вперед проехать это расстояние. Касание коробки не требуется, однако расстояние до нее в момент остановки робота должно быть минимальным.

Пятясь задним ходом вернуться назад, ориентируясь на значение, хранящееся в памяти.

Задача считается выполненной, если робот остановится не дальше 5 см от флажка, но не опрокинет его.

Задача 2

Робот стоит на игровом столе. Так же на столе находятся две одинаковые картонные коробки. Расстояние между коробками не менее 50 см. Робот находится между ним. Расстояние от робота до любой коробки от 5 до 100 см, более точных данных нет.

Задание:

Робот должен указать ближайшую к нему коробку, повернувшись к ней и издав звуковой сигнал.

# Олимпиадные задачи

Задача 1

Робот находится на игровом поле, на котором нарисованы три черные пересекающиеся линии. Толщина каждой линии около 3 см.

Линии являются маршрутами, которые ведут от базы робота (первоначальное положение) к мячику. Мячик находится на подставке собранной согласно инструкции прилагаемой к конструктору Lego Mindstorms. Первоначальное положение робота и мячика неизменны. Форма маршрута, его размеры, известны заранее (рис. 5).

Рис. 5. Поле для робота

Задание:

Робот должен проехать по любому из трех маршрутов к мячику, взять его, и вернуться назад. При движении на базу он должен использовать другой маршрут. Вернувшись на базу, он должен разжать клешни.

Если при движении оба ведущих колеса робота оказываются по одну сторону линии, задача считается нерешенной. На решение задачи отводится не более 7 минут.

Задача 2

Игровое поле представляет собой белую поверхность ограниченную прямоугольным контуром размером 2 на 1,5м. Контур образован черной линией толщиной 3 см (рис. 6).

В центре поля на расстоянии 30 см друг от друга расположены 4 мячика синего и красного цвета (из состава конструктора). Взаиморасположение мячиков разного цвета известно заранее. Каждый мячик находится в центре круга радиусом 5 см. Круг нанесен на поверхность игрового поля, его цвет совпадает с цветом размещенного внутри его мячика.

В каждом углу игрового поля находится треугольная зона красного или синего цвета. Игровое поле не имеет бортиков, мячи являются единственными предметами на поле отражающими сигнал радара робота.

Рис. 6. Поле для робота

Задание:

Робот должен отнести каждый мячик в угол поля соответствующего цвета – красный мячик в угол красного цвета, синий в угол синего цвета. Для предотвращения выкатывания мячика каждый угол огорожен шкивами из набора деталей конструктора.