

Министерство образования и науки Мурманской области  
Государственное автономное учреждение дополнительного образования  
Мурманской области  
«Мурманский областной центр дополнительного образования  
«Лапландия»

ПРИНЯТА  
методическим советом  
Протокол  
от 06.09.2017 № 1

Председатель  О. А. Бережнюк

УТВЕРЖДЕНА  
Приказом ГАУДО МО «МОЦДО  
«Лапландия»  
от 07.09.2017 № 521

Директор  В. Кулаков



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА  
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ  
«ПРОГРАММИРОВАНИЕ РОБОТОВ»

Возраст учащихся: **13-17 лет**  
Срок реализации программы: **3 года**

Автор:  
**Федулеев Александр Александрович,**  
педагог дополнительного образования

Мурманск  
2017

## Пояснительная записка

В школьном курсе информатики вопросы программирования рассматриваются лишь в ознакомительном плане и на это выделяется недостаточное количество часов, как следствие – формальное восприятие учащимися основ современного программирования. Образовательная программа «Программирование роботов» направлена на устранение данного пробела.

Знания, полученные при изучении образовательной программы «Программирование роботов», учащиеся могут использовать для самостоятельного написания программ управления роботами. Кроме этого в процессе обучения у учащихся формируются навыки программирования, представление о профессии программиста. Знания и умения, приобретенные в результате освоения программы «Программирование роботов», являются фундаментом для дальнейшего совершенствования мастерства в области объектно-ориентированного программирования, а также помогут учащимся в дальнейшем обучении в вузах и в профессиональной деятельности.

### Актуальность программы

Актуальность образовательной программы обусловлена тем, что в настоящее время одной из задач современного образования является содействие воспитанию нового поколения, отвечающего по своему уровню развития и образу жизни условиям информационного общества. Для этого обучающимся предлагается освоить язык визуального программирования EV3, которые позволят управлять роботами, занятия помогут найти своё место в современном информационном мире.

### Программа составлена на основе:

- специальной литературы по данному виду технического творчества;
- профессионального опыта педагога.

### Отличия

Дополнительная общеобразовательная программа «Робототехника: конструирование и программирование»	Дополнительная общеобразовательная программа «Программирование роботов»
Изменен возрастной диапазон	
возраст учащихся 10-17 лет	возраст учащихся 13-17 лет
Определена цель программы, исходя из поставленной цели сформулированы новые задачи и ожидаемые результаты	
Создание условий для мотивации, подготовки и профессиональной ориентации школьников для возможного продолжения учебы в ВУЗах и	Удовлетворение образовательных потребностей учащихся в области изучения программирования роботов на базе конструктора Lego EV3.

последующей работы на предприятиях по специальностям, связанным с робототехникой.	
Изменен учебно-тематический план	
	См. «Учебно-тематический план»
Содержание программы дополнено новыми разделами Параллельное выполнение задач. Использование таймеров. Объектно-ориентированное программирование.	
Добавлена система оценки и фиксирования образовательных результатов	
Отсутствует	См. «Система оценки и фиксирования образовательных результатов»

**Вид программы:** общеобразовательная, модифицированная.

**Направленность программы:** техническая.

**Срок реализации программы** 3 года.

**Программа рассчитана** на 216 часа.

**Режим занятий:** 2 раза в неделю по 3 часа.

Режим занятий соответствует санитарно–эпидемиологическим требованиям к учреждениям дополнительного образования детей (санитарно-эпидемиологическим правилам и нормативам СанПиН 2.4.4 3172- 14).

**Форма организации занятий:** групповая.

**Возраст учащихся:** 13-17 лет.

**Количество учащихся:** до 15 человек.

**Состав группы:** постоянный.

**Набор учащихся:** свободный.

В дополнительной общеобразовательной программе предусмотрен летний блок занятий по индивидуальным планам учащихся на летний период времени.

### **Цель программы**

Удовлетворение образовательных потребностей учащихся в области изучения программирования роботов на базе конструктора Lego EV3.

### **Основные задачи**

#### *Обучающие*

- познакомить с назначениями и функциями визуального программирования;
- пропедевтика навыков визуального программирования;
- изучить основные принципы визуального программирования;
- освоить инструментарий современной объектно-визуальной среды;

- сформировать практические навыки работы с интегрированной средой программирования EV3;
- формировать представления о визуальной среде программирования как о многоцелевом и универсальном инструменте познания окружающего мира;
- освоить специальную терминологию;

### *Развивающие*

развивать:

- активное творческое мышление,
  - познавательную и творческую активность.
  - навыки компьютерной грамотности.
  - логическое мышление, умение планировать и предугадывать возможные нестандартные ситуации, возникающие в процессе создания программ;
  - ассоциативную возможность мышления;
- формировать
- системный подход (рассмотрение сложных объектов в виде набора более простых составляющих частей и связей между ними);
  - умения проектирования на основе информационного моделирования объектов и процессов;
  - умения решать принципиально новые задачи, порожденные привнесенным информатикой новым информационным подходом к анализу окружающей деятельности.

### *Воспитательные*

- воспитывать аккуратность, трудолюбие, целеустремленность,
- воспитывать адекватное отношение к личным творческим успехам и успехам других.
- воспитание культуры программирования;

#### **Ожидаемые результаты обучения:**

По окончании **первого** года обучения учащиеся:

*познакомятся* с основными возможностями визуального программирования (алгоритм, ветвление, цикл и пр.)

будут *знать* основы программирования роботов, основные блоки и алгоритмические конструкции на языке EV3

*уметь* создавать несложные программы для управления роботами.

По окончании **второго** года обучения учащиеся:

*познакомятся* с основными алгоритмами для управления роботами на основе данных, полученных от сенсоров на языке RobotC.

будут *знать* методы контроля поведения роботов, управления роботом на основе обратной связи.

**уметь** создавать программы среднего уровня сложности для управления роботами.

По окончании **третьего** года обучения учащиеся:

будут **знать** основные методы и алгоритмы для управления роботом в различной окружающей обстановке

**уметь** создавать сложные программы для управления роботами.

Учащиеся должны знать:

- возможности визуального программирования;
- основные отличия визуального программирования от традиционного;
- основные приемы написания программ-приложений;
- требования к написанию и оформлению программ-приложений;

**Формы диагностики результатов обучения:** наблюдение, проверочные задания, тесты, самостоятельные практические работы, соревнования, конференции, семинары.

**УЧЕБНЫЙ ПЛАН**  
**первого года обучения.**

№ п/п	Разделы	Количество часов		
		Всего	Теоретич.	Практич.
1	Вводное занятие.	1	1	--
2	Основы, принципы визуального программирования.	3	3	--
3	Язык EV3 и его возможности.	6	2	4
4	Создание и сохранение программ.	6	3	3
5	Типы данных, переменные и константы.	12	4	8
6	Операции и выражения.	18	6	12
7	Управляющие операторы.	15	7	8
8	Создание собственных блоков в среде EV3.	30	10	20
9	Параллельное выполнение задач.	20	8	12
10	Использование массивов.	20	8	12
11	Вывод графической информации на экран	20	8	12
12	Использование таймеров.	8	3	5
13	Работа с файлами в среде EV3.	8	3	5
14	Логические операторы.	8	2	6
15	Выполнение итогового творческого проекта.	40	10	30
16	Заключительное занятие.	1	1	
	<b>Всего</b>	<b>216</b>	<b>79</b>	<b>137</b>

**Учебный план второго года обучения.**

№ п/п	Разделы	Количество часов		
		Всего	Теоретич.	Практич.
1	Вводное занятие.	1	1	--
2	Основы, принципы процедурного программирования.	3	3	--
3	Язык RobotC и его возможности.	6	2	4
4	Создание и сохранение программ.	6	3	3
5	Типы данных, переменные и константы.	12	4	8
6	Операции и выражения.	18	6	12
7	Управляющие операторы.	15	7	8
8	Создание собственных процедур и функций в среде RobotC.	30	10	20
9	Параллельное выполнение задач.	20	8	12

10	Использование массивов.	20	8	12
11	Вывод графической информации на экран	20	8	12
12	Использование таймеров.	8	3	5
13	Работа с файлами в RobotC.	8	3	5
14	Логические операторы.	8	2	6
15	Выполнение итогового творческого проекта.	40	10	30
16	Заключительное занятие	1	1	
	<b>Всего</b>	<b>216</b>	<b>79</b>	<b>137</b>

### Учебный план третьего года обучения.

№ п/п	Разделы	Количество часов		
		Всего	Теоретич.	Практич.
1	Вводное занятие.	1	1	--
2	Основы, принципы Объектно ориентированного программирования.	10	4	6
3	Программирование виртуальных роботов	40	10	30
4	Создание виртуального окружения роботов.	60	18	42
5	Взаимодействие робота с окружающими предметами	40	10	30
6	Управление роботом с помощью клавиатуры и мыши	24	6	18
7	Выполнение итогового творческого проекта.	40	10	30
8	Заключительное занятие	1	1	
	<b>Всего</b>	<b>216</b>	<b>60</b>	<b>156</b>

## Содержание программы

### Содержательные линии.

В курсе дополнительной образовательной программы «Программирование роботов» рассматриваются:

- Особенности и возможности визуального программирования;
- Многообразие среды разработки проекта;
- Основы программирования.

Для создания проектов используется программа EV3 для первого года обучения, RobotC для второго года обучения, C++ для третьего года обучения.

### Первый год обучения:

#### 1. Вводное занятие.

Теория: Правила техники безопасности и поведения в кабинете информатики и вычислительной техники. Цель содержания и особенности курса. Краткий обзор современных систем программирования, информационных систем и технологий.

#### 2. Основы, принципы визуального программирования.

Теория: Общая характеристика визуальных систем программирования. Принципы. Структура проекта.

#### 3. Язык EV3 и его возможности.

Теория: Язык EV3. Объектная структура языка EV3. Графические возможности EV3

Практические занятия: Среда: главное окно, окно формы, окно инспектора объектов, окно кода программы. Формы: добавление новой формы, разновидности форм, свойства и события. Компоненты: понятие, виды, размещение на форме, правила задания свойств, общие свойства.

#### 4. Создание и сохранение программ

Теория: Понятие о программе. Принцип построения программы.

Практические занятия: Создание, выполнение и сохранение программ. Обработка ошибок.

#### 5. Типы данных, переменные и константы.

Теория: Числовые данные. Логические значения.. Текстовые данные. Переменные. Массивы. Константы. Комментарии.

Практические занятия: Создание простейших программ.

#### 6. Операции и выражения.

Теория: Выражения в языке EV3. Совместимость типов данных. Оператор присваивания. Арифметические операторы. Логические операторы. Операторы сравнения. Строковые операторы. Приоритеты выполнения операций.

Практические занятия: Создание программ с использованием выражений, оператора присваивания, сравнения, логических и строковых операторов.



### **7. Управляющие операторы.**

Теория: Блоки циклов, Виды циклов. Блоки условий. Блоки множественного выбора.

Практические занятия: Создание программ с использованием управляющих операторов.

### **8. Создание собственных блоков в среде EV3.**

Теория: Основные понятия о блоках EV3. Аргументы. Выполнение блоков. Вызовы процедур и функций. Возврат параметров из функций.

Практические занятия: Создание программ с использованием собственных блоков EV3.

### **9. Параллельное выполнение задач.**

Теория: Понятие о параллельных задачах. Свойства и особенности параллельного выполнения команд. Примеры использования параллельных задач в робототехнике.

Практические занятия: Выполнение задач с использованием параллельного выполнения задач.

### **10. Использование массивов.**

Теория: Объявление массивов. Чтение данных из массивов. Запись данных в массив. Примеры использования массивов в робототехнике. Получение данных о количестве элементов массива.

Практические занятия: Создание программ с использованием массивов.

### **11. Вывод графической информации на экран.**

Теория: Выводы графических примитивов. Объединение данных графики и текста. Использование циклов для вывода информации на экран.

Практические занятия: Создание программ с использованием вывода информации на экран.

### **12. Использование таймеров.**

Теория: Назначение таймеров. Запуск таймера. Выполнение команд по таймеру.

Практические занятия: Проектирование и создание программ, использующих таймеры.

### **13. Работа с файлами в среде EV3.**

Теория: Хранение данных во внешних файлах. Чтение и запись данных из файлов. Поиск файлов. Добавление данных в существующий файл.

Практические занятия: Вывод данных из файла на экран. Использование внешнего файла для хранения параметров выполнения программы.

### **14. Логические операторы.**

Теория: Логические операторы И, ИЛИ, НЕ. Выполнение логических операций с данными. Использование логических операторов для управления поведением робота.

Практические занятия: Создание программ, использующих логические операторы..

### **15. Выполнение итогового творческого проекта**

Выполнение проекта с использованием изученного материала

### **16. Итоговое занятие**

Подведение итогов образовательной программы.

**Второй год обучения:****1. Вводное занятие.**

Теория: Правила техники безопасности и поведения в кабинете информатики и вычислительной техники. Цель содержание и особенности курса. Краткий обзор современных систем программирования, информационных систем и технологий.

**2. Основы, принципы визуального программирования.**

Теория: Общая характеристика визуальных систем программирования. Принципы. Структура проекта.

**3. Язык EV3 и его возможности.**

Теория: Язык EV3. Объектная структура языка EV3. Графические возможности EV3

Практические занятия: Среда: главное окно, окно формы, окно инспектора объектов, окно кода программы. Формы: добавление новой формы, разновидности форм, свойства и события. Компоненты: понятие, виды, размещение на форме, правила задания свойств, общие свойства.

**4. Создание и сохранение программ**

Теория: Понятие о программе. Принцип построения программы.

Практические занятия: Создание, выполнение и сохранение программ. Обработка ошибок.

**5. Типы данных, переменные и константы.**

Теория: Числовые данные. Логические значения.. Текстовые данные. Переменные. Массивы. Константы. Комментарии.

Практические занятия: Создание простейших программ.

**6. Операции и выражения.**

Теория: Выражения в языке EV3. Совместимость типов данных. Оператор присваивания. Арифметические операторы. Логические операторы. Операторы сравнения. Строковые операторы. Приоритеты выполнения операций.

Практические занятия: Создание программ с использованием выражений, оператора присваивания, сравнения, логических и строковых операторов.

**7. Управляющие операторы.**

Теория: Блоки циклов, Виды циклов. Блоки условий. Блоки множественного выбора.

Практические занятия: Создание программ с использованием управляющих операторов.

**8. Создание собственных блоков в среде EV3.**

Теория: Основные понятия о блоках EV3. Аргументы. Выполнение блоков. Вызовы процедур и функций. Возврат параметров из функций.

Практические занятия: Создание программ с использованием собственных блоков EV3.

**9. Параллельное выполнение задач.**

Теория: Понятие о параллельных задачах. Свойства и особенности параллельного выполнения команд. Примеры использования параллельных задач в робототехнике.

Практические занятия: Выполнение задач с использованием параллельного выполнения задач.

### **10. Использование массивов.**

Теория: Объявление массивов. Чтение данных из массивов. Запись данных в массив. Примеры использования массивов в робототехнике. Получение данных о количестве элементов массива.

Практические занятия: Создание программ с использованием массивов.

### **11. Вывод графической информации на экран.**

Теория: Выводы графических примитивов. Объединение данных графики и текста. Использование циклов для вывода информации на экран.

Практические занятия: Создание программ с использованием вывода информации на экран.

### **12. Использование таймеров.**

Теория: Назначение таймеров. Запуск таймера. Выполнение команд по таймеру.

Практические занятия: Проектирование и создание программ, использующих таймеры.

### **13. Работа с файлами в среде EV3.**

Теория: Хранение данных во внешних файлах. Чтение и запись данных из файлов. Поиск файлов. Добавление данных в существующий файл.

Практические занятия: Вывод данных из файла на экран. Использование внешнего файла для хранения параметров выполнения программы.

### **14. Логические операторы.**

Теория: Логические операторы И, ИЛИ, НЕ. Выполнение логических операций с данными. Использование логических операторов для управления поведением робота.

Практические занятия: Создание программ, использующих логические операторы..

### **15. Выполнение итогового творческого проекта**

Выполнение проекта с использованием изученного материала

### **16. Заключительное занятие**

Подведение итогов образовательной программы.

## **Третий год обучения:**

### **1. Вводное занятие.**

Теория: Правила техники безопасности и поведения в кабинете информатики и вычислительной техники. Цель содержания и особенности курса. Краткий обзор современных систем программирования, информационных систем и технологий.

### **2. Основы, принципы Объектно ориентированного программирования.**

Теория: Общая характеристика визуальных систем программирования. Принципы. Структура программы, использующей принципы объектного программирования.

### **3. Программирование виртуальных роботов.**

Теория: Знакомство с программой для создания виртуальных роботов. Основные графические возможности.

Практические занятия: Среда: главное окно, окно формы, окно инспектора объектов, окно кода программы. Формы: добавление новой формы, разновидности форм, свойства и события. Компоненты: понятие, виды, размещение на форме, правила задания свойств, общие свойства.

#### **4. Создание виртуального окружения роботов.**

Теория: Знакомство с программой для создания виртуальных объектов. Виды объектов. Основные свойства объектов. Создание, редактирование и удаление объектов.

Практические занятия: Создание новых объектов окружения, используя стандартные шаблоны. Редактирование объектов. Создание собственных объектов.

#### **5. Взаимодействие робота с окружающими предметами**

Теория: Знакомство с программой для создания виртуальных объектов окружения. Виды объектов. Основные свойства объектов. Создание, редактирование и удаление объектов.

Практические занятия: Создание новых объектов окружения, используя стандартные шаблоны. Редактирование объектов. Изучение возможности взаимодействий роботов с окружающими объектами.

#### **6. Управление роботом с помощью клавиатуры и мыши**

##### **7. Выполнение итогового творческого проекта.**

Практические занятия. Программирование задачи движения робота по взаимодействию с объектами

##### **8. Заключительное занятие.**

Теория.

Подведение итогов работы по программе.

Практика. Конкурс на скорость создания программы для робота.

### Методическое обеспечение программы

Для реализации программы используются следующие формы и методы обучения.

Формы обучения: лекция, практикум, работа со специальной литературой, мини-конференция, обсуждение вариантов решения задачи.

#### Методы обучения:

1. Словесные (указания педагога, объяснение нового материала (лекции), индивидуальная консультация).
2. Практическая работа (задания, тесты, составление алгоритмов и схем, решение задач, наблюдение, проведение экспериментов, работа с литературными источниками).
3. Наблюдение (фото и видеосъемка, проведение замеров).
4. Исследовательский (постановка, проведение и обработка результатов опытов и экспериментов, установление причинно-следственных связей).
5. Проблемного обучения (самостоятельный поиск учащимися ответа на поставленную проблему).
6. Иллюстративно-демонстративные (показ, пример, видеоиллюстрация).

#### Система оценки и фиксирования образовательных результатов

В процессе обучения осуществляется контроль за уровнем сформированности знаний, умений и навыков.

Система контроля за усвоением учащимися программы складывается из следующих элементов: опрос, зачеты, самостоятельные работы, соревнования (где можно определить уровень каждого игрока и команды), конкурсы, тесты. Результаты проверки уровня усвоения программы фиксируются педагогом в специально разработанных листах учебных достижений:

В течении учебного года по определению уровня усвоения программы учащимися осуществляется три диагностических среза:

- **входная диагностика** посредством бесед, анкетирования, тестов, где выясняется начальный уровень знаний, умений и навыков обучающихся, а так же выявляются их творческие способности.
- **промежуточная диагностика** позволяет выявить достигнутый на данном этапе уровень ЗУН учащихся, в соответствии с пройденным материалом программы. Предлагаются контрольные тесты, выполнение практических заданий.
- **итоговая диагностика** проводится в конце учебного года (итоговый показ творческих проектов) и предполагает комплексную проверку образовательных результатов по всем

ключевым направлениям. Данный контроль позволяет проанализировать степень усвоения программы учащимися.

Результаты контроля фиксируются в диагностической карте.

### Диагностика результативности образовательного процесса

#### Система оценки и фиксирования образовательных результатов

В процессе обучения осуществляется контроль за уровнем сформированности знаний, умений и навыков.

Система контроля за усвоением учащимися программы складывается из следующих элементов: опрос, зачеты, самостоятельные работы, соревнования (где можно определить уровень каждого игрока и команды), конкурсы, тесты. Результаты проверки уровня усвоения программы фиксируются педагогом в специально разработанных листах учебных достижений:

В течение учебного года по определению уровня усвоения программы учащимися осуществляется три диагностических среза:

- **входная диагностика** посредством бесед, анкетирования, тестов, где выясняется начальный уровень знаний, умений и навыков учащихся, а так же выявляются их творческие способности.
- **промежуточная диагностика** позволяет выявить достигнутый на данном этапе уровень ЗУН учащихся, в соответствии с пройденным материалом программы. Предлагаются контрольные тесты, выполнение практических заданий.
- **итоговая диагностика** проводится в конце учебного года (итоговый показ творческих проектов) и предполагает комплексную проверку образовательных результатов по всем ключевым направлениям. Данный контроль позволяет проанализировать степень усвоения программы учащимися.

Результаты контроля фиксируются в диагностической карте.

### Виды контроля

Таблица 1

Виды контроля	Содержание	Методы	Сроки контроля
Предварительный	Начальный уровень подготовки учащихся, имеющиеся знания, умения и навыки, связанные с предстоящей	Наблюдение, анкетирование.	Сентябрь

	деятельностью.		
Текущий	Освоение учебного материала по темам.		Октябрь-апрель
Промежуточный	Освоение учебного материала за полугодие	Опросы, практические задания	Декабрь-январь
Итоговый		Конкурс на скорость создания программы для робота. Программирование задачи движения робота по взаимодействию с объектами.	Май

Предварительная диагностика  
по образовательной программе дополнительного образования детей  
«Программирование роботов»

Таблица 2

Наличие первоначальных умений и навыков учащихся, связанных с предстоящей деятельностью:
<ul style="list-style-type: none"><li>• знание названия деталей Lego,</li><li>• основные знания о графическом программировании роботов</li><li>• умение собирать роботов по предложенной схеме,</li><li>• умение пользоваться ПК,</li><li>• умение содержать в порядке рабочее место,</li><li>• умение доводить работу до конца.</li></ul>



Промежуточная диагностика  
по образовательной программе дополнительного образования детей  
«Программирование роботов»

Педагог д/о \_\_\_\_\_  
Группа № \_\_\_\_\_ год обучения \_\_\_\_\_  
Уровень теоретических знаний и / или  
Уровень практических умений и навыков  
Форма проведения \_\_\_\_\_

№ п/п	ФИ Учащегося	Количество баллов
1.		
2.		
3.		
4.		
5.		
6.		
7.		
8.		
9.		
10.		
11.		
12.		
13.		
14.		
15.		

Низкий уровень –

учащийся со значительной помощью педагога ориентируется в содержании учебного материала и дает определение понятиям; освоил отдельные навыки и умения (1-2 балла).

Средний уровень –

почти полное усвоение учебного материала, принимает старательное участие в ответах на вопросы и в заданиях, иногда требуется помощь педагога. Учащийся старателен, внимательно слушает, но ответы нуждаются в уточнении; допускает неточности в работе (3-4 балла).

Высокий уровень –

учащийся самостоятельно ориентируется в содержании пройденного учебного материала, принимает активное участие в ответах на вопросы, полное усвоение содержания учебного материала; способен дать оценку собственной работе, умеет применять теоретические знания и практические умения и навыки в самостоятельной работе (5 баллов).

Средний балл \_\_\_\_\_

## Оценка уровней освоения программы

Таблица 4

Уровни / количество баллов	Параметры	Показатели
Высокий уровень/ 5 баллов	Теоретические знания.	Учащийся освоил материал в полном объеме. Знает и понимает значение терминов, самостоятельно ориентируется в содержании материала по темам. Учащийся заинтересован, проявляет устойчивое внимание к выполнению заданий.
	Практические умения и навыки.	Способен применять практические умения и навыки во время выполнения самостоятельных заданий. Правильно и по назначению применяет инструменты. Работу аккуратно доводит до конца. Может оценить результаты выполнения своего задания и дать оценку работы своего товарища.
	Конструкторские способности.	Учащийся способен узнать и выделить объект (конструкцию, устройство). Учащийся способен собрать объект из готовых частей или построить с помощью инструментов. Учащийся способен выделять составные части объекта. Учащийся способен видоизменить или преобразовать объект по заданным параметрам. Учащийся способен из преобразованного или видоизмененного объекта, или его отдельных частей собрать новый.
Средний уровень/ 3-4 балла	Теоретические знания.	Учащийся освоил базовые знания, ориентируется в содержании материала по темам, иногда обращается за помощью к педагогу. Учащийся заинтересован, но не всегда проявляет устойчивое внимание к выполнению задания.
	Практические умения и навыки.	Владеет базовыми навыками и умениями, но не всегда может выполнить самостоятельное задание, затрудняется и просит помощи педагога. В работе допускает небрежность, делает ошибки, но может устранить их после наводящих вопросов или самостоятельно. Оценить результаты своей деятельности может с подсказкой педагога.
	Конструкторские способности.	Учащийся может узнать и выделить объект (конструкцию, устройство). Учащийся не всегда способен самостоятельно разобрать, выделить составные части конструкции. Учащийся не способен видоизменить или преобразовать объект по заданным параметрам без подсказки педагога.
Низкий уровень / 1-2 балла	Теоретические знания.	Владеет минимальными знаниями, ориентируется в содержании материала по темам только с помощью педагога.
	Практические умения и	Владеет минимальными начальными навыками и умениями. Учащийся способен выполнять каждую операцию только с подсказкой педагога или товарищей. Не всегда правильно применяет необходимый

	навыки.	инструмент или на использует вовсе. В работе допускает грубые ошибки, не может их найти их даже после указания. Не способен самостоятельно оценить результаты своей работы.
	Конструкторские способности.	Учащийся с подсказкой педагога может узнать и выделить объект (конструкцию, устройство). Учащийся с подсказкой педагога способен выделять составные части объекта. Разобрать, выделить составные части конструкции, видоизменить или преобразовать объект по заданным параметрам может только в совместной работе с педагогом.

Сводная таблица результатов обучения  
по образовательной программе дополнительного образования детей  
«Программирование роботов»

**Таблица № 5**

педагог д/о \_\_\_\_\_.

год обучения \_\_\_\_\_

группа № \_\_\_\_\_

№ п/п	ФИ учащегося	Оценка теоретических знаний	Оценка практических умений и навыков	Конструкторские способности	Средний балл
1.					
2.					
3.					
4.					
5.					
6.					
7.					
8.					
9.					
10.					
11.					
12.					
13.					
14.					
15.					

Средний балл \_\_\_\_\_

## Условия реализации программы

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 EV3.

Конструктор LEGO Mindstorms позволяет школьникам в форме познавательной игры узнать многие важные идеи и развить необходимые в дальнейшей жизни навыки. Lego-робот поможет в рамках изучения данной темы понять основы робототехники, наглядно реализовать сложные алгоритмы, рассмотреть вопросы, связанные с автоматизацией производственных процессов и процессов управления. Робот рассматривается в рамках концепции исполнителя, которая используется в курсе информатики при изучении программирования. Однако в отличие от множества традиционных учебных исполнителей, которые помогают учащимся разобраться в довольно сложной теме, Lego-роботы действуют в реальном мире, что не только увеличивает мотивационную составляющую изучаемого материала, но вносит в него исследовательский компонент.

Работает Lego Mindstorms на базе компьютерного контроллера NXT/EV3, который представляет собой двойной микропроцессор, Flash-памяти в каждом из которых более 256 кбайт, Bluetooth-модуль, USB-интерфейс, а также экран из жидких кристаллов, блок батареек, громкоговоритель, порты датчиков и сервоприводов. Именно в NXT/EV3 заложен огромный потенциал возможностей конструктора Lego Mindstorms. Память контроллера содержит программы, которые можно самостоятельно загружать с компьютера. Информацию с компьютера можно передавать как при помощи кабеля USB, так и используя Bluetooth. Кроме того, используя Bluetooth можно осуществлять управление роботом при помощи мобильного телефона.

Для решения поставленных педагогических задач необходим разнообразный дидактический материал и технические средства обучения.

В их числе:

- персональные компьютеры;
- наборы конструкторов Lego NXT или Lego EV3;
- проектор или интерактивная доска;
- наборы специальных полей для соревнований;
- методические разработки мероприятий;

Помещение для проведения учебных и практических занятий должно соответствовать санитарно-эпидемиологическим требованиям. В начале и середине учебного года проводится инструктаж по технике безопасности с целью знакомства с правилами пожарной безопасности, ПДД, поведение на улице, в общественных местах.

В качестве основного педагогического инструмента выступает проверка теоретических знаний на практике.

**Дидактический, раздаточный и демонстрационный материал:**

- тематические папки к разделам программы;
- тестовые задания,
- раздаточный материал;
- иллюстративный материал;
- мультимедийные презентации;
- видеофильмы.

### **Материально-техническое обеспечение программы**

Для реализации дополнительной образовательной программы «Основы робототехники» необходимо иметь:

- На рабочих местах учащихся должны быть обеспечены уровни искусственной освещенности люминесцентными лампами при общем освещении помещений не ниже:  
в учебных помещениях для теоретических занятий - 300 - 500 лк;  
в компьютерных кабинетах - 300 - 500 лк;
- рабочие столы,
- доска демонстрационная,
- шкафы и стеллажи для хранения техники и конструкторов.

#### **Оборудование**

- персональные компьютеры 15 шт.;
- наборы конструкторов LegoNXT или LegoEV3 15 шт.;
- проектор или интерактивная доска 1 шт.;
- наборы специальных полей для соревнований 5 шт.

Медицинская аптечка для оказания доврачебной помощи.

### Список литературы

1. Босова, Л.Л. Информатика: учебник для 6 класса. / Л.Л. Босова. – М.: Бином, 2007. – 208 с.
2. Брага, Н.С. Создание роботов в домашних условиях. / Н.С.Брага. – М.: НТ Пресс, 2007. – 368 с.
3. Гиффорд, К. Роботы. / К. Гиффорд–М.: Росмэн-Пресс, 2009. – 28 с.
4. Жимарши, Ф. Сборка и программирование мобильных роботов в домашних условиях. / Ф. Жимарши. – М.: НТ Пресс, 2007. – 288 с.
5. Индустрия развлечений. ПервоРобот. Книга для учителя и сборник проектов. – М.: LEGO Group, перевод ИНТ. – 87 с.
6. Накано, Э. Введение в робототехнику. / Э. Накано.– М.: Мир, 1988. 329 с.
7. Основы робототехники на базе конструктора LegoMindstorms NXT. // Компьютерные инструменты в школе, 2010.
8. Предко, М. 123 эксперимента по робототехнике. / пер. с англ. Попова В.П. – М.: НТ-Пресс, 2007. – 544с.
9. Юревич, Е.Основы робототехники: учеб.пособие. / Е. Юревич. – 2-е изд. – СПб.: БХВ – Петербург, 2005. – 203 с.

### Список литературы для учащихся

1. Боголюбов, А.Н., Никитин, Д.А. Популярно о робототехнике. / А.Н Боголюбов, Д.А. Никитин. – Киев: Наук.думка, 1989. – 200 с.
2. Горячев, А.В. Информатика в играх и задачах. / А.В. Горячев, К.И Горина, Н.И. Суворова. – М.: Баласс, 2009. – 112 с.
3. Филиппов, С.А Робототехника для детей и родителей. / С.А Филиппов,. – СПб.: Наука, 2010, – 195 с.



## Приложения

**Первичный тест на знание деталей Lego и умение собирать модели роботов по готовой схеме.**



### Задание

1. Необходимо собрать конструкцию робота по прилагаемой схеме.
2. Назвать каждую деталь данной конструкции, ее назначение
3. Найти датчики касания, освещенности, расстояния, звука. Объяснить общий принцип их работы.

## Тест № 1

## «Загрузка прошивки»

Имя \_\_\_\_\_ Дата \_\_\_\_\_

Отметить каждое из следующих утверждений '☑' для Истинного или × для Ложного.

\_\_\_\_\_ Прошивка должна быть загружена каждый раз, перед тем как запускать программы на NXT.

\_\_\_\_\_ Без загруженной прошивки, ваш робот не может выполнять никакие программы.

\_\_\_\_\_, После того, как прошивка ROBOTC загружена, вы сможете выполнять и программы на языке NXT-G и программы на ROBOTC

\_\_\_\_\_ Все прошивки идентичны, после загрузки, вы можете запускать любые программы.

\_\_\_\_\_ Прошивка и программа – это одно и то же.

\_\_\_\_\_ Вы можете загрузить прошивку ROBOTC используя меню Robot – «Compile and Download»

## Тест № 2

## «Загрузка программы»

Имя \_\_\_\_\_ Дата \_\_\_\_\_

Пронумеруйте следующие шаги в том порядке, чтобы вы могли успешно запустить программу. Поместите знак 'X' рядом с шагами, которые не являются обязательными

\_\_\_\_\_ Написать или открыть существующий файл с программой.

\_\_\_\_\_ Нажать серую кнопку на блоке

NXT.

\_\_\_\_\_ Произнести фразу для робота, «Запусти программу».

\_\_\_\_\_ Проверить, что робот включен и подключен кабелем.

\_\_\_\_\_ Выбрать пункт Try Me в меню, используя экран NXT и кнопки.

\_\_\_\_\_ Нажать кнопку Start в окне Program Debug.

\_\_\_\_\_ Выбрать пункт меню Robot – «Compile and Download».

## Тест № 3

## «Программирование в ROBOTC»

Имя \_\_\_\_\_ Дата \_\_\_\_\_

1. Какой символ сигнализирует о конце команды?

\_\_\_\_\_

\_\_\_\_\_

2. Приведите примеры использования парных скобок.

\_\_\_\_\_

\_\_\_\_\_

3. Управляющие структуры, такие как главная задача или блок условия:

- a. Создают проблемы при попытке управлять двигателями
- b. Используются только для удобного чтения программ и игнорируются роботом
- c. Управляют ходом выполнения команд: они выбирают, какие команды и когда выполнятся
- d. Используют парные скобки в своем составе

## Тест № 4

## «Движение вперед»

Имя \_\_\_\_\_ Дата \_\_\_\_\_

1. В этой программе, команда на какой строке контролирует, как долго робот будет двигаться?

```
1 task main()
2 {
3     motor[motorC] = 100;
4     motor[motorB] = 100;
5     wait1Msec(2000);
6 }
```

a. Строка

1

b. Строки 3 и

4

c. Строка

5

d. Робот будет двигаться

бесконечно

2. Взгляните на программу. Напишите ниже код, который заставит робота двигаться со скоростью в 2 раза меньше, но на такое же расстояние.

```
1 task main()
2 {
3     motor[motorC] = 100;
4     motor[motorB] = 100;
5     wait1Msec(2000);
6
7
8
9
10
11
12
13 }
```

## Тест № 5

## «Скорость и Направление»

Имя \_\_\_\_\_ Дата \_\_\_\_\_

1. В следующей программе для того, что бы изменить скорость мотора С, необходимо внести изменения в строку(и):

```
1 task main()
2 {
3     motor[motorC] = 100;
4     motor[motorB] = 100;
5     wait1Msec(4000);
6 }
```

- a. Строка 3
- b. Строка 4
- c. Строка 5
- d. Скорость мотора С нельзя изменить

2. Составьте программу, которая будет выполнять последовательно следующие действия:

Двигаться вперед на 50% мощности 3 секунды.

Повернуться на месте вправо в течении 0,5 секунды на любой скорости.

Ехать назад на максимальной скорости 1 секунду.

```
1 taskmain()
2 {
3
4
5
6
7
8
```

9  
10  
11  
12  
13  
14 }

**Тест № 6****«Более точное движение»**

Имя \_\_\_\_\_ Дата \_\_\_\_\_

1. Какие факторы влияют на то, что робот не может ехать ровно?

- a. Неточности при изготовлении двигателей
- b. Неравномерное распределение веса робота
- c. Различная сила трения внутри шестеренок робота
- d. Все из вышеперечисленного

2. Контроль «с обратной связью» описывает систему:

- a. контролирует собственную скорость и регулирует ее мощность, чтобы достигнуть желаемого результата.
- b. когда характеристики системы являются тайной.
- c. система, в которой в цикле используются открытые и закрытые скобки
- d. системы, в которых используются циклы

3. Команда `nSyncedTurnRatio=100`; задаст режим для ведомого мотора:

- a. на такой же скорости и таком же направлении, что и ведущий мотор
- b. на такой же скорости в противоположном направлении, что и ведущий мотор
- c. на 100 градусов в секунду в таком же направлении, что и ведущий мотор
- d. на полной мощности вперед.

4. PID алгоритм

настраивает:

- a. Мощность каждого мотора так, что бы они вращались с одинаковой скоростью.
- b. Делает мощность обоих моторов одинаковой.
- c. Устанавливает передаточное число на шестеренках мотора.



d. Изменяет силу трения внутри мотора, что позволяет мотору двигаться более ровно.

**5.** Напишите фрагмент кода, который установит режим синхронизации между моторами В и С, в котором С будет главным (ведущим), а мотор В ведомым

1

2

3

4

## Тест № 7

## «Определение стен»

Имя \_\_\_\_\_ Дата \_\_\_\_\_

1. Рекомендуемый способ настройки сенсоров в ROBOTC с помощью диалога Motors and Sensors показанный выше

- a. Да
- b. Нет

2. В какой форме сенсор касания возвращает значения?

- a. Число от 0 до 255, в зависимости от того, как сильно нажат сенсор.
- b. Число 0 или 1, означающее нажат сенсор или нет
- c. Сила нажатия на сенсор.
- d. А или В, в зависимости от того, как настроен сенсор

3. Опишите что будет выполняться в данной программе

```
1 while(SensorValue(touchSensor) == 1)
2 {
3     motor[motorC] = 100;
4     motor[motorB] = 0;
5 }
```

4. В каких случаях выполняются команды внутри цикла?

- a. Когда условие цикла истинно.
- b. Когда условие цикла ложно.
- c. Когда сенсор возвращает какое то значение

## Тест № 8

## «Сенсор расстояния»

Имя \_\_\_\_\_ Дата \_\_\_\_\_

1. Сенсор расстояния использует звук, что бы определить:
  - a. Направление
  - b. Форму
  - c. Температуру
  - d. Расстояние
  
2. В ROBOTC, команда \_\_\_\_\_ используется чтобы получить значение с сенсора.
  - a. SensorDistance()
  - b. SensorValue()
  - c. Sensor()
  - d. SensorMeasurement()
  
3. В программе на ROBOTC не нужно указывать в каких единицах возвращать значение расстояние, потому что сенсор расстояние всегда возвращает значение в
  - a. Дюймах.
  - b. Сантиметрах.
  - c. Дробных величинах.
  - d. Целых величинах.
  
4. Сенсор расстояния отправляет и принимает звуковые волны, для того что бы определить расстояние до объекта
  - a. Да
  - b. Нет

## Тест № 9

## «Использование переменных»

Имя \_\_\_\_\_ Дата \_\_\_\_\_

1. Если вы хотите сохранить в переменную `my_variable` дробное значение,

то тип этой переменной должен быть

\_\_\_\_\_

2. Что обозначает "объявить"  
переменную?

\_\_\_\_\_

\_\_\_\_\_

3. Какие из этих вариантов НЕ могут использоваться в качестве имени переменных.

a. true

b. my\_variable

c. var1x

d. ants go marching

e. 1\_by\_1

f. one\_by\_one

g. motorh.

PB&amp;J

4. Используя этот код, добавьте строку которая запишет в переменную "product" значение равное произведению «a» на «b». Какое значение будет в переменной "product" ?

1 int a;

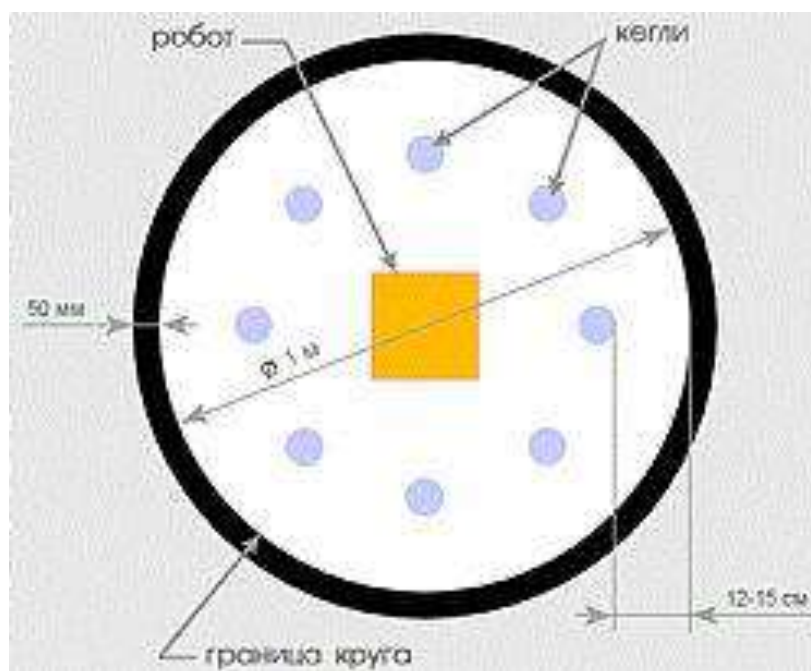
2 int b;

3 int product;

4 a= 10;

5 b= 100;

## Задание «Кегельринг ВВ»



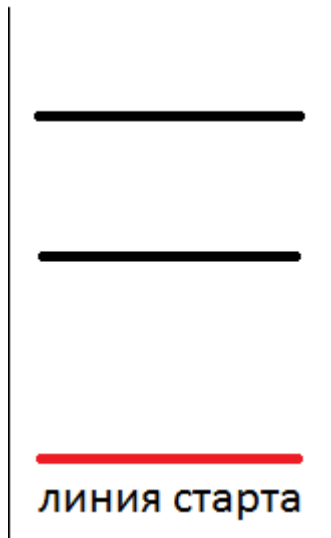
На поле устанавливаются 4 черных и 4 белых банки

**Необходимо собрать конструкцию робота, имеющего в своем составе:**

1. Простую тележку, используя два мотора и поддерживающее колесо.
2. Датчик освещенности, способный определить наличие черной линии на поле.
3. Датчик освещенности, способный определить цвет банки
4. Возможно использование датчика расстояния, что бы определять наличие банок.

**Необходимо написать программу для выполнения следующего задания**

1. Робот начинает движение из центрального квадрата.
2. Задача робота, выбить все кегли белого цвета за пределы черного круга.
3. Черные банки должны остаться внутри круга.

**Контрольная работа по теме****«Программирование роботов»**

**Необходимо собрать конструкцию робота, имеющего в своем составе:**

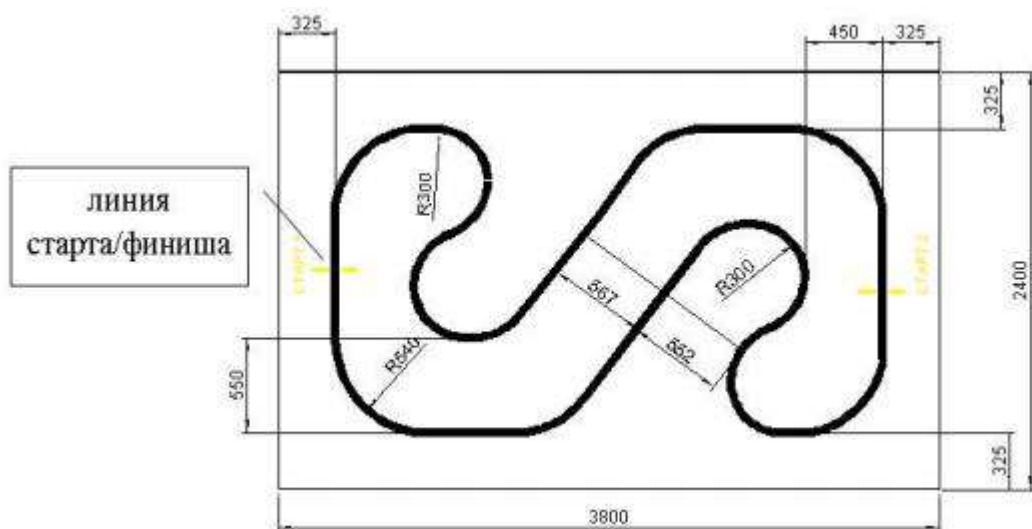
1. Простую тележку, используя два мотора и поддерживающее колесо.
2. Датчик касания, находящийся впереди робота для определения стены.
3. Датчик освещенности, способный определить наличие черных линий на поле.

**Необходимо написать программу для выполнения следующего задания**

1. Робот стартует с красной линии и двигается вперед, пока не коснется стены.
2. Во время движения робот должен подсчитать количество черных линий и расстояние от точки старта до каждой из линий
3. Информацию о расстоянии до каждой из черных линий необходимо вывести на экран.
4. Робот должен вернуться обратно на линию старта, используя значения датчика оборотов двигателей.

## Контрольная работа по теме

## «Основы теории автоматического управления»



**Необходимо собрать конструкцию робота, имеющего в своем составе:**

1. Простую тележку, используя два мотора и поддерживающее колесо.
2. Один или два датчика освещенности для определения черной линии на поле.

Максимальная ширина робота 250 мм, длина 250 мм.

Высота и вес робота не ограничены.

Робот должен быть автономным (не допускается дистанционное управление роботом)

**Необходимо написать программу для выполнения следующего задания**

Используя любой из алгоритмов теории автоматического управления, робот должен преодолеть всю дистанцию быстрее 35 секунд.

1. Учащиеся разбиваются на пары и выбирают тему для творческого проекта.
2. Вместе с преподавателем необходимо определить цель проекта, его задачи, какие функции будет выполнять данный робот и что ему для этого потребуется.
3. Необходимо разработать конструкцию робота, создать для него программу.
4. Для представления проекта необходимо создать презентацию и подготовить рассказ.