

**Министерство образования и науки Республики Татарстан  
Муниципальное бюджетное учреждение дополнительного образования  
«Центр детского технического творчества»  
Елабужского муниципального района Республики Татарстан**

Принята на заседании  
педагогического совета  
от «28» сентября 20\_\_ г.  
Протокол № 2



**Дополнительная  
общеобразовательная общеразвивающая программа  
технической направленности**

**«3 D Технологии»**

Возраст обучающихся: 13-17 лет  
Срок реализации: 2 года

Автор-составитель:  
**Минкин Александр Владимирович,**  
педагог дополнительного образования,  
первая квалификационная категория

г. Елабуга, 2017 г.

### Информационная карта образовательной программы

<b>1.</b>	<b>Образовательная организация</b>	Муниципальное бюджетное учреждение дополнительного образования «Центр детского технического творчества» Елабужского муниципального района Республики Татарстан
<b>2</b>	<b>Полное название программы</b>	Дополнительная общеобразовательная общеразвивающая программа технической направленности «3D технологии»
<b>3</b>	<b>Направленность программы</b>	техническая
<b>4</b>	<b>Сведения о разработчиках</b>	
4.1	Педагог дополнительного образования	Минкин Александр Владимирович
<b>5</b>	<b>Сведения о программе</b>	
5.1	Срок реализации	2 года
5.2	Возраст обучающихся	13-17 лет
5.3	Характеристика программы: - тип программы - вид программы - принцип проектирования программы - форма организации содержания и учебного процесса	Дополнительная общеобразовательная программа общеразвивающая
5.4	Цель программы	Научить использовать технические средства и средства информационных технологий, чтобы проводить исследования и решать задачи в межпредметной деятельности.
5.5	Образовательные модули ( в соответствии с уровнями сложности содержания и материала программы)	
<b>6</b>	<b>Формы и методы образовательной деятельности</b>	Традиционные и нетрадиционные формы организации учебной деятельности. Методы: наглядные, словесные, игровые, практические
<b>7</b>	<b>Формы мониторинга результативности</b>	Теоретическая, практическая подготовка, общеучебные умения и навыки, личностное развитие ребенка
<b>8</b>	<b>Результативность реализации программы</b>	Аттестация учащихся, результаты мониторинга
<b>9</b>	<b>Дата утверждения и последней корректировки программы</b>	28 сентября 2017 года
<b>10</b>	<b>Рецензенты</b>	Рецензент, доцент ЕИ КФУ, к.п.н. Дерягин А.В.

## **ОГЛАВЛЕНИЕ**

1. Пояснительная записка
2. Учебно-тематический план
3. Содержание программы
4. Организационно-педагогические условия реализации программы
5. Формы аттестации, мониторинг результатов
6. Список литературы
7. Календарный учебный график (Приложение №1)

## **1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

### **1.1 Направленность программы**

Программа по содержательной направленности – техническая; по функциональному предназначению – учебно-познавательная. Программа предназначена для подростков 13-17 лет, обучающихся в основной школе, и направлена на обеспечение дополнительной подготовки по технической направленности (программирование, информатика, инженерия, робототехника).

### **1.2. Обоснование необходимости внедрения**

Основным содержанием данного курса являются постепенное усложнение занятий от технического моделирования до сборки и программирования роботов.

Актуальность курса заключается в том, что он направлен на формирование творческой личности живущей в современном мире. Технологические наборы LEGO MINDSTORMS NXT 2.0 и Arduino ориентированы на изучение основных физических принципов и базовых технических решений, лежащих в основе всех современных конструкций и устройств.

На уроках используются конструктор “Базовый набор 8547” серии LEGO MINDSTORMS NXT 2.0 с программным обеспечением ПервоРобот (CD-R диск с визуальной средой программирования NXT-G) и комплект Arduino.

Используя персональный компьютер, либо нетбук или ноутбук с ПО NXT-G, LEGO-элементы из конструктора ученики могут конструировать управляемые модели роботов. Загружая управляющую программу в специальный LEGO-компьютер NXT и присоединяя его к модели робота, робот функционирует автономно. NXT работает независимо от настольного компьютера, на котором была написана управляющая программа; получая информацию от различных датчиков и обрабатывая ее, он управляет работой моторов.

Итоги изученных тем подводятся созданием учениками собственных автоматизированных моделей, с написанными для данной модели программами, и защитой этих проектов в устной форме и в соревновании (которое можно провести в классе).

### **1.3. Отличительные особенности предлагаемой дополнительной образовательной программы от уже существующих.**

Используется идея компетентностно-ориентированного образования и деятельностный подход обучения. Основной акцент в программе сделан на обучение через практику, продуктивную работу учащихся в малых группах, использование межпредметных связей, развитие самостоятельности учащихся и личной ответственности за принятие решений.

Идея формирующего оценивания как наиболее адекватного требованиям современного общества механизма саморегуляции образовательного процесса, учитывающего личностные особенности учащихся, содействующего выработке у обучающихся способности к самооценке, стимулирующей их образовательную активность

### **1.4. Цель и задачи программы, формирование УУД**

**Цель:** Научить использовать технические средства и средства информационных технологий, чтобы проводить исследования и решать задачи в междисциплинарной деятельности.

**Задачи:**

**Обучающие**

1. Знакомство со средой программирования NXT-G, Arduino;
2. Изучение основ программирования (составление алгоритмов, программ);
3. Проектирование роботов и программирование их действий;

**Воспитательные**

1. Через создание собственных проектов проследить пользу применения роботов в реальной жизни;
2. Расширение области знаний о профессиях;
3. Практическое применение в технической сфере человеческой деятельности;

**Развивающие**

1. Формирование интереса к освоению познавательной, творческой деятельности;
2. Умение учеников работать в группах.

**Требования к уровню освоения содержания дисциплины.**

Учащийся, изучивший дисциплину, должен знать:

1. Языки программирования: NXT-G, Arduino;
2. Основные датчики и устройства сопряжения с компьютером;
3. Архитектуру микроконтроллера.

Учащийся должен уметь:

1. Программировать на языке: NXT-G, Arduino;
2. Создавать простейшие схемы и устройства для автоматизированного управления каким-либо узлом или устройством;

Учащийся должен владеть навыками:

1. проведения программно-аппаратного эксперимента для решения разных задач;
2. численных расчетов для решения задач автоматизации;
3. представления информации различными способами (в вербальной, знаковой, аналитической, математической, графической, схемотехнической, образной, алгоритмической формах).

**В результате освоения программы идет формирование следующих УУД:**

**Личностные:**

1. Развитие устойчивой учебно-познавательной мотивации учения;
2. Развитие профессиональной – компетентности в решении проблем, основанных на собственном выборе;
3. Формирование целостного мировоззрения, соответствующего уровню развития современной техники;

4. Формирование коммуникативной компетентности в общении со сверстниками (осознанного и уважительного отношения к участнику группы и его мнению, освоение норм, правил поведения и ролей в группе);
5. Развитие опыта практической деятельности, который пригодится в жизненно важных ситуациях.

#### Регулятивные действия:

1. Умение самостоятельно определить цель проекта;
2. Умение спланировать свою и коллективную деятельность для более эффективного решения поставленных задач;
3. Умение оценивать правильность выполнения задачи и разнообразие возможных решений;
4. Умение принимать правильные решения в короткие сроки;
5. Умение оценить свою работу и работу других участников коллектива.

#### Познавательные действия:

1. Поиск и выделение необходимой информации, в том числе решение рабочих задач с использованием инструментов ИКТ и дополнительных источников информации;
2. Умение сопоставлять функциональные устройства NXT и их характеристики;
3. Умение обобщать и классифицировать рассматриваемые объекты (компоненты для компьютера);
4. Умение самостоятельно определить критерии для определения функциональных возможностей некоторых программных блоков NXT и устанавливать причинно-следственные связи;
5. Умение строить логические рассуждения и делать выводы применительно к поставленным задачам.

#### Коммуникативные действия:

1. Умение организовать совместную деятельность со сверстниками и учителем;
2. Умение работать в группе и находить общее решение;
3. Умение отстаивать свои интересы и интересы группы;
4. Умение формулировать и аргументировать свою точку зрения;
5. Умение осознанно использовать речевые средства для выражения своих чувств, мыслей и потребностей в соответствии с задачами и условиями коммуникации.

#### **1.5. Возраст детей, участвующих в реализации программы**

Возраст детей 8-10 класс общеобразовательной школы (13-17 лет).

### **1.6. Сроки реализации программы**

Программа рассчитана на 2 года обучения с общим объемом 288 ч., по 144 часа в год

### **1.7. Форма и режим занятий.**

Общий объем занятий 144 ч. в год. Занятия проводятся два раза в неделю по два часа каждое занятие. Занятия проходят в форме лекций, на которых учащиеся знакомятся с теоретической стороной вопроса и в форме лабораторных (практических) занятий, на которых они самостоятельно (индивидуально или группой) решают конкретно поставленные задачи. Занятия проходят в компьютерном классе (20 компьютеров) с использованием специального оборудования (конструктор Lego Mindstorms NXT и Arduino)

### **1.8. Ожидаемые результаты реализации программы и способы их проверки**

#### **Личностные**

В программе курса предусмотрено ознакомление обучающихся с историческими фактами об открытиях российских ученых, о выдающихся достижениях отечественной науки, влияющих на развитие робототехники. Описываются общемировые тенденции развития робототехники.

Рассматриваются вопросы правового регулирования в интеллектуальной сфере. Объясняются положения федерального закона о защите интеллектуальной собственности на практических примерах (жизненных ситуациях). Проводится работа по разъяснению актуальных вопросов правового регулирования в сфере робототехники.

Объяснение теоретических основ робототехники производится с опорой на материальную базу и актуальные представления в развитии робототехники, при этом используется современная аппаратная, математическая и программная база. Рассмотрение вопросов и проблем проходит в контексте мировых соревнований по робототехнике (WRO).

Задания выполняются индивидуально и в группах по (2-3 человека). Работа проводится фронтально. Задания носят практический характер, например создание робота способного преодолевать препятствия высотой 12,5 см. Созданные обучающимися работы (проекты) публично демонстрируются (защищаются), роботы участвуют в соревновании (в классе).

В процессе создания и во время защиты роботов (проектов), поднимается ряд вопросов, связанных с анализом изучаемого материала, необходимостью аргументированного доказательства «жизнеспособности» своей конструкции.

## **Предметные**

Формируется представление о роли робототехники в обществе. Обучающиеся овладевают системой базовых знаний, отражающих вклад робототехники в формирование современной научной картины мира. В процессе изучения основ программирования роботов, обучающиеся овладевают навыками алгоритмического мышления и понимают необходимость формального описания алгоритмов. Овладевают понятием сложности алгоритма. Например, можно построить робота, который будет ездить по линии используя простой алгоритм «зиг-заг»-го образного движения. Однако, демонстрация более сложного ПИД – алгоритма дает выигрыш в скорости, при этом сложность алгоритма на порядок выше. На занятиях обучающиеся учатся понимать программы, написанные на графическом языке программирования (NXT) и языке программирования высокого уровня (ARDUINO); знакомятся с основными конструкциями и элементами программирования, учатся использовать основные управляющие конструкции. Овладевают навыками и опытом разработки программ в среде программирования (NXT, ARDUINO), включая тестирование и отладку программ; учатся навыкам формализации прикладной задачи и документирования программ, интерпретации результатов, получаемых в ходе моделирования реальных процессов. Например, опыты по измерению температуры тела (смесь воды и льда) или по измерению ускорения свободного падения, дают возможность реализовать проекты, связанные с постройкой робота и интерпретации результатов. При работе с оборудованием необходимо соблюдение требований техники безопасности, гигиены и ресурсосбережения.



### **Метапредметные**

В процессе освоения курса по робототехнике, обучающиеся могут самостоятельно определять цели и составлять план работы по созданию роботов. Например, при создании модели робота преодолевающего препятствия на трассе, педагог только отмечает общий круг проблем, который необходимо рассмотреть при создании робота (общий алгоритм движения, основные элементы конструкции). Однако, модели создаются разными группами обучающихся (а так же индивидуально), поэтому приходится каждому самостоятельно ставить цель и план для постройки эффективной модели робота. При этом педагог должен контролировать и корректировать научно-исследовательскую деятельность обучающихся; помогать им использовать всевозможные ресурсы для достижения целей. Важно, чтобы обучающиеся научились продуктивно общаться и взаимодействовать в процессе совместной деятельности, могли учитывать позиции другого, эффективно разрешать конфликты.

Необходимо создать условия к самостоятельному поиску методов решения практических задач. В некоторых случаях требуется статистический подход к решению задачи. Например, в задаче для робота «езда по линии» подбор коэффициентов, есть статистическая задача. Но для эффективного решения обучающимся приходится прибегнуть и к самостоятельному поиску информации в различных источниках (книги, журналы, Интернет), при этом важно критически оценивать и интерпретировать информацию, получаемую из различных источников.

#### **1.9. Формы подведения итогов реализации программы**

Итоговая аттестация проводится в форме соревнования.

## 2. УЧЕБНО - ТЕМАТИЧЕСКИЙ ПЛАН

### 1-й год обучения

№	Тема	Количество часов		
		теор.	практ	всего
1.	Введение в робототехнику. Конструкторы компании ЛЕГО. Знакомимся с набором Lego Mindstorms NXT 2.0 сборки 8547	2		2
2.	Калибровка датчика. Шины данных. Шины данных - Повреждение	2		2
3.	Мультизагрузка NXT. Файлы и память устройства NXT	2		2
4.	Профили. Цепочка программы	2		2
5.	Простой текст . Горячие клавиши NXT	2		2
6.	Начальная точка. Обновление операционной системы NXT	2		2
7.	Создание Проекта	2		2
8.	Блок «Экран». Блок «Цикл»	2		2
9.	Блок «Движение». Блок «Писать/Играть»	2		2
10.	Блок «Звук» . Блок «Переключатель»	2		2
11.	Блок «Ожидание». Блок «Экран»	2		2
12.	Блок «Лампа». Блок «Мотор»	2		2
13.	Блок «Отправить сообщение»	2		2
14.	Блок «Звук»	2		2
15.	Блок «Датчик освещённости»	2		2
16.	Блок «Кнопки NXT»	2		2
17.	Блок «Приём сообщений»	2		2
18.	Блок «Датчик оборотов». Блок «Датчик звука»	2		2
19.	Блок «Датчик температуры».		2	2
20.	Блок «Таймер»		2	2
21.	Блок «Датчик касания»		2	2
22.	Блок «Датчик расстояния»		2	2
23.	Блок «Цикл»		2	2
24.	Блок «Стоп». Блок «Переключатель»		2	2
25.	Блок «Ожидание». Блок «Сравнение»		2	2
26.	Блок «Логика». Блок «Математика»		2	2
27.	Блок «Случайное число». Блок «Интервал»		2	2
28.	Блок «Переменная». Блок «Константа»		2	2
29.	Блок «Калибровка». Блок «Доступ к файлам»		2	2
30.	Блок «Не засыпать». Блок «Число в текст»		2	2
31.	Блок «Обнулить мотор». Блок «Текст»		2	2

32.	Блок «Bluetooth соединение». Блок «Начать регистрацию данных»		2	2
33.	Блок «Остановить регистрацию данных». Анализ данных		2	2
34.	Панель параметров эксперимента. Пульт регистрации данных		2	2
35.	Таблица данных. График		2	2
36.	Управление файлами экспериментов. Создание прогноза. Удаление данных		2	2
37.	Программа. Сборка. Класс	2		2
38.	Система типов		2	2
39.	Управляющие операторы и методы	2		2
40.	Объекты		2	2
41.	Массивы	2		2
42.	Перегруженные операции		2	2
43.	Наследование и полиморфизм	2		2
44.	Интерфейсы		2	2
45.	Делегаты и события	2		
46.	Атрибуты, сборки, рефлексия		2	2
47.	Ввод/вывод	2		2
48.	Коллекции. Параметризованные классы		2	2
49.	Шаблоны	2		2
50.	Совмещение управляемого и неуправляемого кодов		2	2
51.	Потоки	2		2
52.	Форма		2	2
53.	GDI+	2		2
54.	Основы ADO .NET		2	2
55.	Основные понятия	2		2
56.	Обработчики событий: onMouseover		2	2
57.	Запросы пользователю и переменные	2		2
58.	Концепция свойств		2	2
59.	Создание функций	2		2
60.	Команды последствия: onUnLoad и onMouseOut		2	2
61.	Открываем новые окна	2		2
62.	Открытие окна с помощью функции		2	2
63.	Метод 'Confirm' (Введение в if и else)	2		2
64.	Математические вычисления		2	2
65.	Изменение изображения с помощью события onMouseOver	2		2

66.	Вызов функции в формы		2	2
67.	Передача данных в функцию	2		2
68.	Оператор if и ветвление		2	2
69.	Случайный выбор фраз и изображений	2		2
70.	Введение в циклы		2	2
71.	Массивы	2		2
72.	Слайд-шоу		2	2
	<b>Всего:</b>	<b>72</b>	<b>72</b>	<b>144</b>

## 2-й год обучения

№	Тема	Количество часов		
		теор.	практ	всего
1.	Программирование Arduino - введение	2		2
2.	Структура программы, константы	2		2
3.	Цифровой ввод/вывод	2		2
4.	Аналоговый ввод/вывод	2		2
5.	Дополнительные функции ввода/вывода	2		2
6.	Работа со временем	2		2
7.	Математические функции	2		2
8.	Псевдослучайные числа	2		2
9.	Последовательная передача данных	2		2
10.	Прерывания	2		2
11.	EEPROM	2		2
12.	Создание своей библиотеки	2		2
13.	Blink без delay	2		2
14.	Создание библиотеки. Ethernet library – библиотека для работы с Ethernet	2		2
15.	Создание библиотеки. Servo library – библиотека для работы с сервомашинками	2		2
16.	Создание библиотеки. Firmata library – библиотека, реализующая протокол Firmata	2		2
17.	Создание библиотеки. OneWireSlave - библиотека эмуляции устройства 1-Wire	2		2
18.	Arduino/CraftDuino и PROTEUS. Arduino/CraftDuino и WinAVR - программируем на чистом C	2		2
19.	Цифровой ввод - кнопка		2	2
20.	Аналоговый вывод - Fading		2	2
21.	Аналоговый ввод – потенциометр		2	2
22.	Аналоговый ввод – осциллограф		2	2
23.	Генерация звука – пьезоизлучатель.		2	2
24.	Фоторезистор		2	2

25.	Сенсор на светодиоде		2	2
26.	Общение с Arduino - программирование работы с COM-портом.		2	2
27.	Аналоговый датчик температуры – LM335		2	2
28.	Подключаем к Arduino мышку PS/2		2	2
29.	Протокол 1-Wire и iButton		2	2
30.	Arduino и эмулятор iButton		2	2
31.	Arduino и температурный 1-Wire датчик DS18S20		2	2
32.	Arduino и драйвер двигателей L293D (Простой мотор-шилд)		2	2
33.	Arduino и сервомашинка		2	2
34.	Как с помощью Arduino/CraftDuino можно управлять устройствами на 220В		2	2
35.	Подключаем LCD-дисплей на базе HD44780 к Arduino		2	2
36.	ИК-датчик препятствий для Arduino на базе фототранзистора		2	2
37.	Что такое ASP .NET	2		2
38.	Анатомия ASP .NET. ASP .NET в действии		2	2
39.	Серверные элементы управления	2		2
40.	Серверные элементы управления (продолжение)		2	2
41.	AutoPostBack. Привязка к данным. Коллекции. Проверка правильности вводимых данных	2		2
42.	<u>Работа с базами данных</u>		2	2
43.	Работа с базами данных (продолжение). Элементы-источники данных (Data Source Controls)	2		2
44.	Элементы-потребители данных. Data-Bound Controls		2	2
45.	Элементы-потребители данных ASP .NET 2.0	2		
46.	Работа с XML		2	2
47.	Навигация по сайту	2		2
48.	Шаблоны дизайна страниц ASP .NET		2	2
49.	Стили элементов управления. Темы и шкурки	2		2
50.	Пользовательские и собственные серверные элементы управления		2	2
51.	Конфигурирование проекта. Глобальные объекты	2		2
52.	Web-сервисы		2	2
53.	Локализация проектов. Аутентификация пользователей	2		2

54.	Элементы управления и использование JavaScript		2	2
55.	Введение в PHP	2		2
56.	Области применения и способы использования. Создание первого скрипта на PHP. Основы синтаксиса		2	2
57.	Управляющие конструкции	2		2
58.	Обработка запросов с помощью PHP		2	2
59.	Характеристика методов Post и Get.	2		2
60.	Функции в PHP		2	2
61.	Объекты и классы в PHP	2		2
62.	Работа с массивами данных		2	2
63.	Работа со строками	2		2
64.	Работа с файловой системой		2	2
65.	Базы данных и СУБД. Введение в SQL	2		2
66.	Взаимодействие PHP и MySQL		2	2
67.	Авторизация доступа с помощью сессий	2		2
68.	Настройка сессий в файлах php.ini, httpd.conf, .htaccess.		2	2
69.	Регулярные выражения	2		2
70.	Взаимодействие PHP и XML		2	2
71.	Использование шаблонов в PHP	2		2
72.	Проект на PHP.		2	2
	Всего:	72	72	144

### 3. СОДЕРЖАНИЕ ПРОГРАММЫ

#### 1-й год обучения

1. Введение в робототехнику. Конструкторы компании ЛЕГО. Знакомимся с набором Lego Mindstorms NXT 2.0 сборки 8547
2. Калибровка датчика. Шины данных. Шины данных - Повреждение
3. Мультизагрузка NXT. Файлы и память устройства NXT
4. Профили. Цепочка программы
5. Простой текст . Горячие клавиши NXT
6. Начальная точка. Обновление операционной системы NXT
7. Создание Проекта
8. Блок «Экран». Блок «Цикл»
9. Блок «Движение». Блок «Писать/Играть»
10. Блок «Звук» . Блок «Переключатель»
11. Блок «Ожидание». Блок «Экран»
12. Блок «Лампа». Блок «Мотор»
13. Блок «Отправить сообщение»

14. Блок «Звук»
15. Блок «Датчик освещённости»
16. Блок «Кнопки NXT»
17. Блок «Приём сообщений»
18. Блок «Датчик оборотов». Блок «Датчик звука»
19. Блок «Датчик температуры».
20. Блок «Таймер»
21. Блок «Датчик касания»
22. Блок «Датчик расстояния»
23. Блок «Цикл»
24. Блок «Стоп». Блок «Переключатель»
25. Блок «Ожидание». Блок «Сравнение»
26. Блок «Логика». Блок «Математика»
27. Блок «Случайное число». Блок «Интервал»
28. Блок «Переменная». Блок «Константа»
29. Блок «Калибровка». Блок «Доступ к файлам»
30. Блок «Не засыпать». Блок «Число в текст»
31. Блок «Обнулить мотор». Блок «Текст»
32. Блок «Bluetooth соединение». Блок «Начать регистрацию данных»
33. Блок «Остановить регистрацию данных». Анализ данных
34. Панель параметров эксперимента. Пульс регистрации данных
35. Таблица данных. График
36. Управление файлами экспериментов. Создание прогноза. Удаление данных

### 37. Программа. Сборка. Класс

В данной лекции рассматриваются основные понятия языка, а также принципы работы Microsoft .NET

### 38. Система типов

Язык программирования предполагает наличие правил построения корректных предложений. В свою очередь, предложения строятся из выражений. Основной характеристикой выражения является значение этого выражения. Можно утверждать, что выполнение программы состоит из вычисления значений выражений, которые образуют предложения программы. Основы синтаксиса C# изложены в этой лекции

### 39. Управляющие операторы и методы

В языке программирования C# существуют специальные операторы, которые в зависимости от вычисляемых значений выражений позволяют управлять ходом выполнения программы, эта лекция рассказывает именно о них

### 40. Объекты

В соответствии с принципами объектно-ориентированного программирования решение поставленной задачи сводится к разработке модели (объявлению класса) и созданию экземпляров (объектов), представляющих реализацию этой модели. В этой лекции обсуждаются проблемы, связанные с созданием и последующим уничтожением объектов

#### 41. Массивы

В этой лекции обсуждаются массивы. Функциональность класса массива, категории массивов, синтаксис объявления, инициализация, применение массивов

#### 42. Перегруженные операции

Основная конструкция С# – объявление класса. Класс есть тип. Тип характеризуется неизменяемым набором свойств и методов. Для предопределенных типов определены множества операций, которые кодируются с использованием множества определенных в языке символов операций. Мощнейшее средство языка С# перегрузка операций обсуждается в этой лекции

#### 43. Наследование и полиморфизм

Наследование и полиморфизм являются одними из принципов ООП. О том как они реализованы в С# в этой лекции

#### 44. Интерфейсы

Интерфейсы фактически те же самые абстрактные классы, не содержащие объявлений данных-членов и объявлений обычных функций. О них рассказано в этой лекции.

#### 45. Делегаты и события

Класс, структура, интерфейс, перечисление, делегат – это все разнообразные категории классов. Каждая категория имеет свои особенности объявления, свое назначение и строго определенную область применения. Об особенностях делегатов в данной лекции

#### 46. Атрибуты, сборки, рефлексия

Рефлексия представляет собой процесс анализа типов (структуры типов) в ходе выполнения приложения (сборки). В .NET рефлексия реализуется свойствами и методами класса System.Type и классов пространства имен System.Reflection

#### 47. Ввод/вывод

Применительно к обсуждаемым проблемам ввода/вывода в программах на С#, поток – это последовательность байтов, связанная с конкретными устройствами компьютера (дисками, дисплеями, принтерами, клавиатурами) посредством системы ввода/вывода. Подробнее о потоках - в этой лекции

#### 48. Коллекции. Параметризованные классы

Списки, очереди, двоичные массивы, хэш-таблицы, словари – все это коллекции. Существуют различные типы (классы) коллекций. Объект – представитель данного класса коллекции характеризуется множеством функциональных признаков, определяющих способы работы с элементами (неважно какого типа), которые образуют данную коллекцию

#### 49. Шаблоны

В данной лекции рассказывается о мощном средстве языка С# - шаблонах. Шаблонный класс (и функция) обеспечивают стандартную реализацию какой-либо функциональности для подстановочного класса

#### 50. Совмещение управляемого и неуправляемого кодов



.NET появилась не на пустом месте. Вновь разрабатываемый управляемый код вынужден взаимодействовать с существующим неуправляемым программным кодом. Поэтому на платформе .NET предусмотрены различные сценарии установления взаимодействия между управляемым и неуправляемым кодами. О них рассказано в этой лекции

#### 51. Потоки

В этой лекции рассказывается о принципах управления потоками в языке C#

#### 52. Форма

В этой лекции рассказывается о формах. Форма — это класс. Форма предназначена для реализации интерфейса пользователя приложения. Содержит большой набор свойств, методов, событий для реализации различных вариантов пользовательского интерфейса. Является окном и наследует классу Control

#### 53. GDI+

GDI+ (Graphic Device Interface+ — Интерфейс Графических Устройств) — это подсистема Microsoft Windows XP, обеспечивающая вывод графической информации на экраны и принтеры. GDI+ является преемником GDI, интерфейса графических устройств, включаемого в ранние версии Windows. Интерфейс GDI+ изолирует приложение от особенностей конкретного графического оборудования. Такая изоляция позволяет разработчикам создавать аппаратно-независимые приложения. Взаимодействию GDI и C# посвящена эта лекция.

#### 54. Основы ADO .NET

ADO .NET (ActiveX Data Objects .NET) является набором классов, реализующих программные интерфейсы для облегчения подключения к базам данных из приложения независимо от особенностей реализации конкретной системы управления базами данных и от структуры самой базы данных, а также независимо от места расположения этой самой базы — в частности, в распределенной среде (клиент-серверное приложение) на стороне сервера. Работу C# с ADO обсуждает данная лекция

#### 55. Основные понятия

Введение в JavaScript. Размещение сценария на Web-странице. Ошибки в сценариях. Виды ошибок и методы их поиска. Функции для работы с датой и временем.

#### 56. Обработчики событий: onMouseover

События в JavaScript. Обработка события onMouseOver. Описание основных событий и методы их обработки.

#### 57. Запросы пользователю и переменные

Работа с переменными. Запросы пользователю.

#### 58. Концепция свойств

Объектная модель документа. Основные объекты и их свойства. Структура объектов JavaScript, порядок их подчинения.

#### 59. Создание функций

Функции в JavaScript. Их создание и использование.

60. Команды последствия: onUnLoad и onMouseOut  
Использование событий onUnLoad и onMouseOut
61. Открываем новые окна  
Методы создания и работы с новыми окнами.
62. Открытие окна с помощью функции  
Основные функции для манипуляции с окнами.
63. Метод 'Confirm' (Введение в if и else)  
Основы логического разветвления сценариев.
64. Математические вычисления  
Математические функции и их применение.
65. Изменение изображения с помощью события onMouseOver  
Работа с изображениями. Динамическое изменение изображений.  
Динамическое изменение изображений с помощью функций.
66. Вызов функции в формы  
Создание функций для обработки данных пользователя в формах.  
Динамическое изменение содержимого форм.
67. Передача данных в функцию  
Базовые аспекты обработки данных функциями. Методы создания случайных событий и работа с ними.
68. Оператор if и ветвление  
Логическое разветвление программ. Условный оператор.  
Примеры работы с условиями.
69. Случайный выбор фраз и изображений  
Выбор и обработка случайных данных.
70. Введение в циклы  
Повторяющиеся действия. Циклы с предусловием. Циклы с постусловием.
71. Массивы  
Введение в массивы. Их создание и обработка.
72. Слайд-шоу  
Динамическая загрузка и обновление изображений. Анимация. Основы мультипликации в JavaScript. Предварительная обработка данных форм.

## **2-й год обучения**

1. Программирование Arduino – введение  
Знакомство и начало работы с Arduino Uno: о том что это и зачем она нужна; о подключении к компьютеру и установке программной среды на Windows XP, Windows 7;
2. Структура программы, константы  
Знакомство со средой программирования, структурой программы, используемые типы переменных и написание первой программы для Arduino.
3. Цифровой ввод/вывод  
Подключении кнопок к Arduino Uno; о том зачем нужен breadboard. Объясняется как применяется широтно-импульсная модуляция (также известная как ШИМ или PWM). Рассматривается возможность создавать

собственные функции в среде программирования Arduino на примере функции стабилизации неустойчивого сигнала.

#### 4. Аналоговый ввод/вывод

Использование аналоговых входов Arduino. С помощью схемы делителя напряжения и фоторезистора собираем фотосенсор. На основе него делаем прототип устройства резервного освещения. И возможности применения инфракрасного дальномера.

#### 5. Дополнительные функции ввода/вывода

Рассматриваются базовые понятия схемотехники. О токе и напряжении, о том как применяются резисторы и как строятся делители напряжения. О стягивающих, подтягивающих и токоограничивающих резисторах. О законе Ома. О сенсорах и регуляторах напряжения. О подстроечных резисторах и аналоговых входах. И о том, как всё это соотносится с Arduino.

#### 6. Работа со временем

Рассматриваются дополнительные специфичные функции Arduino для работы со временем `unsigned long millis(void)`; Вызов: `time = millis()`;

#### 7. Математические функции

Рассматриваются специфичные математические функции Arduino: `min(x, y)`, `max(x, y)`, `abs(x)`, `constrain(x, a, b)`, `map(value, fromLow, fromHigh, toLow, toHigh)`, `pow(base, exponent)`, `sq(x)`, `sqrt(x)`, `sin(rad)`, `cos(rad)`, `tan(rad)`.

#### 8. Псевдослучайные числа

Рассматриваются специфичные функции Arduino: `void randomSeed(unsigned int seed)`.

#### 9. Последовательная передача данных

Рассматривается Arduino/Freduino встроенный контроллер для последовательной передачи данных, который может использоваться как для связи между Arduino/Freduino устройствами, так и для связи с компьютером. На компьютере соответствующее соединение представлено либо обычным COM-портом (в случае Arduino Single-Sided Serial Board), либо USB COM-портом, который появляется в системе после установки необходимого драйвера.

#### 10. Прерывания

Знакомство с прерываниями. Прерывание (англ. `interrupt`) — сигнал, сообщающий процессору о наступлении какого-либо события. При этом выполнение текущей последовательности команд приостанавливается, и управление передаётся обработчику прерывания, который выполняет работу по обработке события и возвращает управление в прерванный код.

#### 11. EEPROM

Знакомство с EEPROM. EEPROM — (Electrically Erasable Programmable Read-Only Memory) электрически стираемое перепрограммируемое ПЗУ, ЭСППЗУ). Память такого типа может стираться и заполняться данными несколько десятков тысяч раз. Используется в твердотельных накопителях. Одной из разновидностей EEPROM является флеш-память (*Flash Memory*).

#### 12. Создание своей библиотеки

На примере библиотеки для работы с PS/2-устройствами оценивается работа через библиотеку. Библиотека – это дополнительный класс, который инкапсулирует в себе функции для работы с устройством, т.о. для создания библиотеки нужно создать один .h файл в котором будет описываться класс/функции/ константы и один .cpp, в котором все эти функции реализуются :)

### 13. Blink без delay

Пример использования millis() для «одновременного» выполнения нескольких задач.

14. Создание библиотеки. Ethernet library – библиотека для работы с Ethernet Ethernet-шилд – это дополнительная плата, которая подключается к ардуино и позволяет этой замечательной плате посылать и получать данные через локальную сеть и даже работать Интернетом. Работа с протоколами TCP/IP в шилде реализована аппаратно на микросхеме W5100.

15. Создание библиотеки. Servo library – библиотека для работы с сервомашинками

Servo library – библиотека для работы с сервомашинками. Сервомашинка – это мотор-редуктор, с обратной связью, благодаря которой можно повернуть выходной вал на строго определённый угол. Стандартные рулевые машинки, вертятся от 0 до 180 градусов (также есть сервы постоянного вращения – т.н. Continuous Rotation Servo, чуть позже покажем, как обычную серву можно переделать до такого состояния).

16. Создание библиотеки. Firmata library – библиотека, реализующая протокол Firmata

Библиотека Firmata реализует протокол Firmata, что позволяет простым образом общаться с программами на компьютере. Данная библиотека входит в состав Arduino IDE. Удобна тем, что при необходимости не нужно изобретать велосипед и придумывать свой протокол, а уже использовать этот готовый протокол.

17. Создание библиотеки. OneWireSlave - библиотека эмуляции устройства 1-Wire

Библиотека OneWireSlave позволяет использовать ардуину на линии 1-Wire в качестве ведомого устройства. Теоретически на линии может быть до  $2^{48}$  или 281 триллионов ведомых устройств. С помощью этой библиотеки на одну линию совместно с датчиками температуры и прочими устройствами, использующими протокол 1-Wire, можно подключить несколько ардуин.

18. Arduino/CraftDuino и PROTEUS. Arduino/CraftDuino и WinAVR - программируем на чистом C

— PROTEUS может симулировать несколько МК в одном устройстве (AVR, PIC, Motorola, ARM)

— в поставку PROTEUS входит очень много моделей компонентов.

— можно симулировать любые электронные устройства и не содержащие МК

На Arduino/CraftDuino/Freeduino можно программировать не только на Wiring, но и на других языках программирования.

Это можно сделать на обычном C/C++. Для этого используется компилятор **WinAVR**. WinAVR — это интегрированная среда разработки; она включает мощные компиляторы Си и ассемблера, программатор AVRDUDE, отладчик, симулятор и множество других вспомогательных программ и утилит.

#### 19. Цифровой ввод – кнопка

Рассмотрим цифровой ввод. Возьмём обычную кнопку и соберём простую схему:

#### 20. Аналоговый вывод – Fading

Рассмотрим скетч Fading (Sketchbook – Examples – Analog — Fading ), как пример работы с функцией analogWrite — аналоговым выводом сигнала. Значение, которое можно передавать в analogWrite должно быть между 0 и 255, что соответствует напряжению от 0V до 5V на выходе порта. Т.о. постепенно увеличивая/уменьшая переданное в analogWrite значение можно добиться плавного роста/падения напряжения на выходе. Подключив к одному из PWM-портов ардуины (**3, 5, 6, 9, 10, 11**, а на платах на базе ATmega8 только — **9, 10, 11**) светодиод будет плавно загораться и затухать (fading).

#### 21. Аналоговый ввод – потенциометр

Рассмотрим аналоговый ввод — для этого используем переменный резистор, подключённый к аналоговому входу.

#### 22. Аналоговый ввод – осциллограф

У Arduino есть такая замечательная функция — analogRead(), которая считывает значение с указанного аналогового порта. Значит мы можем получить простенький осциллограф. Аналоговый ввод – осциллограф. Для этого будем считывать данные с аналогового порта Arduino/Freedom и записывать их в последовательный (COM) порт. А уже из последовательно порта их будет принимать наша программа и строить график сигнала.

#### 23. Генерация звука – пьезоизлучатель.

Пьезокерамические излучатели (пьезоизлучатели) — электроакустические устройства воспроизведения звука, использующие пьезоэлектрический эффект (эффект возникновения поляризации диэлектрика под действием механических напряжений (прямой пьезоэлектрический эффект). Существует и обратный пьезоэлектрический эффект — возникновение механических деформаций под действием электрического поля.

#### 24. Фоторезистор

Фоторезистор — полупроводниковый прибор, изменяющий величину своего сопротивления при облучении светом.

#### 25. Сенсор на светодиоде

Рассмотрено нестандартное использование светодиода.

#### 26. Общение с Arduino - программирование работы с COM-портом.

Рассматривается в пример консольная POSIX C – программа — программа может принимать и передавать данные на плату Arduino/Freedom

#### 27. Аналоговый датчик температуры – LM335

Рассмотрен вариант использования терморезистора. Рассмотрим, как работать с аналоговыми датчиками температуры, на примере **LM335**.

## 28. Подключаем к Arduino мышку PS/2

Рассмотрим работу с устройствами PS/2.

## 29. Протокол 1-Wire и iButton

Знакомство с 1-Wire и работа с устройством, использующим этот протокол.

## 30. Arduino и эмулятор iButton

Рассматривается Arduino и эмулятор iButton

## 31. Arduino и температурный 1-Wire датчик DS18S20

DS18S20 – высокоточный цифровой термометр с интерфейсом 1-Wire (High Precision 1-Wire Digital Thermometer) от DALLAS Semiconductor (Maxim). Для подключения датчика достаточно двух проводов – линии данных и заземления.

## 32. Arduino и драйвер двигателей L293D (Простой мотор-шилд)

Используется популярный драйвер для управления моторчиками – L293D. L293D содержит два драйвера для управления электродвигателями небольшой мощности. Имеет две пары входов для управляющих сигналов и две пары выходов для подключения электромоторов. Кроме того, у L293D есть два входа для включения каждого из драйверов. Эти входы используются для управления скоростью вращения электромоторов с помощью ШИМ (PWM).

## 33. Arduino и сервомашинка

Рассматривается мотор-редуктор, способный поворачивать выходной вал строго в заданное положение (на угол) и удерживать его там, вопреки сопротивлениям и возмущениям недружелюбной среды.

## 34. Как с помощью Arduino/CraftDuino можно управлять устройствами на 220В

Рассматривается управление высоковольтной нагрузкой с помощью ардуины, достаточно подавать управляющий сигнал от ардуины на базу ключевых транзисторов, которые управляют включением/выключением реле.

## 35. Подключаем LCD-дисплей на базе HD44780 к Arduino

Рассматривается LCD-дисплей на базе контроллера HD44780 на примере WH1602B-YYK-СТК.

## 36. ИК-датчик препятствий для Arduino на базе фототранзистора

Рассматривается аналоговый сенсор на фототранзисторе. Так как датчик аналоговый, то его выход должен подключаться к аналоговым портам контроллера Arduino (на вход АЦП микроконтроллера). По величине аналогового сигнала мы сможем примерно оценивать расстояние до препятствия (разумеется, абсолютных величин мы получить не сможем, так как уровень сигнала будет меняться в зависимости от объекта). Простейшая схема — это пара из ИК-светодиода и фототранзистора:

## 37. Что такое ASP .NET

Дается обзор различных web-технологий. Объясняется преимущество ASP.NET. Описывается процесс подготовки среды разработки приложений. Создается первая динамическая страница. Вводится понятие структуры проекта.

## 38. Анатомия ASP .NET. ASP .NET в действии

Представление двух этапов компиляции. Знакомство с просмотрщиком классов. Директивы и их атрибуты. Способы внедрения кода ASP .NET в страницу. Пример двуязычного проекта.

#### 39. Серверные элементы управления

Дается обзор классов элементов управления ASP .NET и их сравнение с HTML. Работа с дизайнером форм.

#### 40. Серверные элементы управления (продолжение)

Рассмотрены элементы управления — кнопки и изображения, календарь.

#### 41. AutoPostBack. Привязка к данным. Коллекции. Проверка правильности вводимых данных

Привязка к данным с помощью коллекций. Примеры на использование коллекций ArrayList, Hashtable. Элементы управления группы Validation.

#### 42. Работа с базами данных

Дается представление о базах данных. Хранимые процедуры сервера SQL. Что такое ADO. Окно внешних источников данных. Создание страниц данных.

#### 43. Работа с базами данных (продолжение). Элементы-источники данных (Data Source Controls)

Описание нового типа элементов управления группы Data.

#### 44. Элементы-потребители данных. Data-Bound Controls

Элементы управления для отображения данных.

#### 45. Элементы-потребители данных ASP .NET 2.0

Элементы управления для отображения данных ASP .NET 2.0.

#### 46. Работа с XML

Краткая аннотация: XML как универсальный язык данных. Чтение и запись в XML. Файлы преобразования и схемы документа. Элементы управления, отображающие XML-данные.

#### 47. Навигация по сайту

Описание элементов управления группы Navigation. Навигация по сайту. Отображение иерархической информации.

#### 48. Шаблоны дизайна страниц ASP .NET

Описывается тип страниц Master Page, программное управление шаблонами дизайна, вложенные шаблоны дизайна, последовательность событий страницы при наличии Master Page.

#### 49. Стили элементов управления. Темы и шкурки

Стили элементов управления. Внешние файлы стиля. Темы и шкурки.

#### 50. Пользовательские и собственные серверные элементы управления

Как создать свой собственный элемент управления.

#### 51. Конфигурирование проекта. Глобальные объекты

Конфигурационные файлы, переменные события приложения и сессии, работа с состоянием отображения.

#### 52. Web-сервисы

Представление о сервисах. Понятие о SOAP. Создание web-сервиса. Ресурсы Интернета с готовыми сервисами.

#### 53. Локализация проектов. Аутентификация пользователей

Локализация проектов. Модель безопасности ASP.NET 2.0. Роли и членство. Аутентификация и авторизация. Профили пользователей.

#### 54. Элементы управления и использование JavaScript

Некоторые элементы управления ASP .NET 2.0. Использование JavaScript, механизм обратного вызова.

#### 55. Введение в PHP

Лекция содержит: историю языка (от PHP/FI до PHP5); описание его возможностей (краткий перечень платформ, протоколов, баз данных, приложений электронной коммерции и функций, которые поддерживаются PHP); области применения (как серверное приложение, в командной строке, создание GUI приложений); способы использования (как PHP встраивается в HTML-код и простейшие примеры.); создание первого скрипта на PHP. Установка и настройка программного обеспечения, необходимого для работы с PHP.

#### 56. Основы синтаксиса

Лекция посвящена изучению основ синтаксиса PHP. Рассматриваются способы разделения инструкций, создания комментариев, переменные, константы и типы данных, операторы. Пример – создание заготовки письма.

#### 57. Управляющие конструкции

В лекции рассматриваются условные операторы (if, switch), работа с циклами (while, for, foreach) и использование функций include, require. Пример – универсализация письма в зависимости от ситуации и его отправка каждому из группы пользователей.

#### 58. Обработка запросов с помощью PHP

Лекция посвящена изучению способов отправки данных на сервер и их обработке с помощью PHP. Рассматриваются основные понятия клиент-серверных технологий. Рассматривается понятие HTML-формы и отправка данных с ее помощью. Дается краткая

#### 59. Характеристика методов Post и Get.

Рассматривается механизм получения данных из HTML-форм и их обработка с помощью PHP. Пример – создание формы для регистрации пользователей на сайте, отправка "универсального письма" всем зарегистрировавшимся.

#### 60. Функции в PHP

В лекции рассматриваются: понятие функции, функции, определяемые пользователем, аргументы функций, передача аргументов по значению и по ссылке, значение аргументов по умолчанию и значения, возвращаемые функцией (функция return()). Пример – создание web-интерфейса для генерации HTML-формы.

#### 61. Объекты и классы в PHP

Понятия класса и объекта. Определение и использование классов. Понятие расширения класса. Конструкторы. Оператор :: Базовый класс и функция parent. Пример – автоматическая генерация по желанию пользователя представителей классов статей или личностей, а также их отображения на странице браузера.

#### 62. Работа с массивами данных



Лекция посвящена более подробному изучению массивов и функций, встроенных в PHP для работы с ними. Рассматриваются функции для поиска элементов в массиве, для сортировки элементов массива, а также применение созданных пользователем функций ко всем элементам массива, разбивка массива на подмассивы и другое

#### 63. Работа со строками

В данной лекции более подробно обсуждаются вопросы работы со строками, изучаются функции, полезные для решения разнообразных прикладных задач. Рассматриваются различные способы вывода строк, разбивка и соединение строк (функции `explode`, `implode`), определение длины строки (`strlen`), выделение подстроки (`strstr`, `substr`).

#### 64. Работа с файловой системой

В лекции обсуждаются вопросы, связанные с созданием файлов, чтением данных из файла, удалением файла, а также проверка наличия файла на сервере. (Функции `fopen`, `fwrite`, `fclose`, `file`, `fget`, `unlink`, `file_exists`.) Пример - загрузка файла на сервер с помощью web-интерфейса.

#### 65. Базы данных и СУБД. Введение в SQL

В лекции рассматриваются понятия базы данных и СУБД, дается краткое описание существующих типов баз данных (сетевые, реляционные, иерархические). Рассматриваются основы языка запросов SQL: операции выбора, добавления, изменения и удаления строки, а также операции создания, изменения и удаления таблицы. База данных MySQL. Использование PhpMyAdmin для взаимодействия с базой данных MySQL. Обсуждаются основные принципы отображения объектной модели документа на реляционную структуру базы данных. Пример – проектирование базы данных виртуального музея истории.

#### 66. Взаимодействие PHP и MySQL

Лекция предназначена для знакомства со способами взаимодействия PHP и СУБД MySQL. Основное внимание уделяется установке соединения с базой данных, функциям отправки запросов и обработке ответов (`mysql_connect`, `mysql_query`, `mysql_result`, `mysql_num_rows`, `mysql_close`). Пример - создание web-интерфейса для администрирования базы данных виртуального музея истории.

#### 67. Авторизация доступа с помощью сессий

Лекция посвящена изучению вопросов обеспечения безопасности в сети и использованию для этих целей механизма сессий. Рассматриваются: инициализация сессий, передача идентификатора пользователя, регистрация переменных сессии, уничтожение сессии.

#### 68. Настройка сессий в файлах `php.ini`, `httpd.conf`, `.htaccess`.

Приводится пример авторизации пользователя с помощью механизма сессий.

#### 69. Регулярные выражения

В лекции рассматривается понятие регулярного выражения, реализация механизма регулярных выражений в языке PHP, их синтаксис и семантика. Пример – задача контекстного поиска.

## 70. Взаимодействие PHP и XML

Лекция знакомит с понятием объектной модели XML-документа и ее использованием в PHP. Установка расширения DOM XML. Обработка элементов XML документа с помощью функций PHP (получение значения узла, атрибута и т.п.) Пример – использование XML-базы данных в качестве альтернативы реляционной СУБД (реализация административного интерфейса каталога виртуального музея).

## 71. Использование шаблонов в PHP

В лекции рассматривается понятие шаблона и его использование в языке программирования PHP, а также классы шаблонов FastTemplate и Smarty. В качестве примера использования различных типов шаблонов и их обработчиков решается задача генерации списка документов по таблице базы данных.

## 72. Проект на PHP.

# 4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

## Практическая работа (Arduino)

### Введение

Arduino программируется на специальном языке программирования – он основан на C/C ++, и позволяет использовать любые его функции. Строго говоря, отдельного языка Arduino не существует, как и не существует компилятора Arduino – написанные программы преобразуются (с минимальными изменениям) в программу на языке C/C++, и затем компилируются компилятором AVR-GCC. Так что фактически, используется специализированный для микроконтроллеров AVR вариант C/C++.

Разница заключается в том, что Вы получаете простую среду разработки, и набор базовых библиотек, упрощающих доступ к находящейся «на борту» микроконтроллера периферии.

Согласитесь, очень удобно начать работу с последовательным портом на скорости 9600 бит в секунду, сделав вызов одной строчкой:

```
Serial.begin(9600);
```

А при использовании «голого» C/C++ Вам бы пришлось разбираться с документацией на микроконтроллер, и вызывать нечто подобное:

```
UBRR0H = ((F_CPU / 16 + 9600 / 2) / 9600 - 1) >> 8;
```

```
UBRR0L = ((F_CPU / 16 + 9600 / 2) / 9600 - 1);
```

```
sbi(UCSR0B, RXEN0);
```

```
sbi(UCSR0B, TXEN0);
```

```
sbi(UCSR0B, RXCIE0);
```

Здесь кратко рассмотрены основные функции и особенности программирования Arduino. Если Вы не знакомы с синтаксисом языков C/C++, советуем обратиться к любой литературе по данному вопросу, либо Internet-источникам.

С другой стороны, все представленные примеры очень просты, и скорее всего у Вас не возникнет трудностей с пониманием исходных текстов и

написанием собственных программ даже без чтения дополнительной литературы.

Более полная документация (на английском языке) представлена на официальном сайте проекта – <http://www.arduino.cc> . Там же есть форум, ссылки на дополнительные библиотеки и их описание.

По аналогии с описанием на официальном сайте проекта Arduino, под «портом» понимается контакт микроконтроллера, выведенный на разъем под соответствующим номером. Кроме того, существует порт последовательной передачи данных (СОМ-порт).

### **Структура программы**

В своей программе Вы должны объявить две основных функции: `setup()` и `loop()`.

Функция `setup()` вызывается один раз, после каждого включения питания или сброса платы Arduino. Используйте её, чтобы инициализировать переменные, установить режимы работы цифровых портов и т.д.

Функция `loop()` последовательно раз за разом исполняет команды, которые описаны в ее теле. Т.е. после завершения функции снова произойдет ее вызов.

Разберем простой пример:

```
void setup()                                // начальные установки
{
  beginSerial(9600); // установка скорости работы серийного порта на 9600 бит/с
  ек
  pinMode(3, INPUT); // установка 3-его порта на ввод данных
}

// Программа проверяет 3-
// ий порт на наличие на нём сигнала и посылает ответ в
// виде текстового сообщения на последовательный порт компьютера
void loop() // тело программы
{
  if (digitalRead(3) == HIGH) // условие на опрос 3го порта
    serialWrite('H');          // отправка сообщения в виде буквы «H» на СОМ-порт
  else
    serialWrite('L');          // отправка сообщения в виде буквы «L» на СОМ-порт
  delay(1000);                 // задержка 1 сек.
}
```

### **Константы**

Константы – предопределенные значения. Они используются, чтобы делать программы более легкими для чтения. Мы классифицируем константы в группах.

#### **Уровни сигналов порта HIGH и LOW**

При чтении или записи к цифровому порту применимо только два возможных значения – порт может быть установлен как HIGH (высокий уровень) или LOW (низкий уровень).

Уровень HIGH соответствует 5 вольтам на выходе. При чтении значения на цифровом порте, начиная с 3 вольт и выше, микропроцессор воспримет это напряжение как HIGH. Эта константа представлена целым числом 1.

Уровень LOW соответствует 0 вольтам на выходе порта. При чтении значения на цифровом порте, начиная с 2 вольт и меньше, микропроцессор воспримет это напряжение как LOW. Эта константа представлена целым числом 0.

Таким образом, оба следующих вызова будут эквивалентны:

```
digitalWrite(13, HIGH); // можно так,  
digitalWrite(13, 1);    // а можно и так
```

Считается, что первый вариант более нагляден.

### **Настройка цифровых портов на ввод (INPUT) и вывод (OUTPUT) сигналов**

Цифровые порты могут использоваться на ввод или вывод сигналов.

Изменение порта с ввода на вывод производится при помощи функции `pinMode()`.

Порты, сконфигурированные на ввод сигналов, имеют большое входное сопротивление, что позволяет подключать к ним источник сигнала, и порт не будет потреблять большой ток.

Порты, сконфигурированные на вывод сигналов, имеют малое выходное сопротивление. Это означает, что такие порты могут обеспечивать подключенные к ним элементы электроэнергией. В этом состоянии порты поддерживают положительное или отрицательное направление тока до 40 мА (миллиампер) на другие устройства или схемы. Это позволяет подключить к ним какую-либо нагрузку, например светодиод (через резистор, ограничивающий ток). Порты, сконфигурированные как выводы, могут быть повреждены, если их замкнуть накоротко на «землю» (общая шина питания), на источник питания +5 В, или подсоединить к мощной нагрузке с малым сопротивлением.

Пример:

```
pinMode(13, OUTPUT); //13й вывод будет выходом  
pinMode(12, INPUT);  //а 12й – входом
```

### **Специфичные для Arduino функции и объекты**

#### **Цифровой ввод/вывод**

##### **pinMode**

Вызов:

```
pinMode (порт, режим);
```

Описание:

Конфигурирует указанный порт на ввод или вывод сигнала.

Параметры:

порт – номер порта, режим которого Вы желает установить (значение целого типа от 0 до 13).

режим – либо INPUT (ввод) либо OUTPUT (вывод).

Пример:

```
pinMode(13, OUTPUT); //13й вывод будет выходом
pinMode(12, INPUT);  //а 12й – входом
```

Примечание:

Аналоговые входы могут использоваться как цифровые входы/выходы, при обращении к ним по номерам с 14 (аналоговый вход 0) по 19 (аналоговый вход 5)

### **digitalWrite**

Вызов:

```
digitalWrite(порт, значение);
```

Описание:

Устанавливает высокий (HIGH) или низкий (LOW) уровень напряжения на указанном порту.

Параметры:

порт: номер порта

значение: HIGH или LOW

Пример:

```
digitalWrite(13, HIGH); // выставляем 13й вывод в «высокое» состояние
```

### **digitalRead**

Вызов:

```
value = digitalRead (порт);
```

Описание:

Считывает значение на указанном порту

Параметры:

порт: номер опрашиваемого порта

Возвращаемое значение: возвращает текущее значение на порту (HIGH или LOW) типа int

Пример:

```
int val;
val = digitalRead(12); // опрашиваем 12й вывод
```

Примечание:

Если к считываемому порту ничего не подключено, то функция digitalRead () может беспорядочно возвращать значения HIGH или LOW.

### **Аналоговый ввод/вывод сигнала**

#### **analogRead**

Вызов:

```
value = analogRead(порт);
```

Описание:

Считывает значение с указанного аналогового порта. Arduino содержит 6 каналов, аналого-цифрового преобразователя на 10 битов каждый. Это означает, что входное напряжения от 0 до 5В преобразовывается в

целочисленное значение от 0 до 1023. Разрешающая способность считывания составляет:  $5 \text{ В} / 1024 \text{ значений} = 0,004883 \text{ В/значение}$  (4,883 мВ). Требуется приблизительно 100 нС (0.0001 С), чтобы считать значение аналогового ввода, так что максимальная скорость считывания - приблизительно 10000 раз в секунду.

Параметры:

порт: номер опрашиваемого аналогового входа

Возвращаемое значение: возвращает число типа `int` в диапазоне от 0 до 1023, считанное с указанного порта.

Пример:

```
int val;
```

```
val = analogRead(0); // считываем значение на 0м аналоговом входе
```

Примечание:

Аналоговые порты по умолчанию определены на ввод сигнала и в отличие от цифровых портов их не требуется конфигурировать с помощью вызова функции `pinMode`.

### **analogWrite**

Вызов:

```
analogWrite(порт, значение);
```

Описание:

Выводит на порт аналоговое значение. Эта функция работает на: 3, 5, 6, 9, 10, и 11 цифровых портах Arduino.

Может применяться для изменения яркости светодиода, для управления двигателем и т.д. После вызова функции `analogWrite`, соответствующий порт начинает работать в режиме широтно-импульсного модулирования напряжения до тех пор, пока не будет следующего вызова функции `analogWrite` (или функций `digitalRead` / `digitalWrite` на том же самом порте).

Параметры:

порт: номер опрашиваемого аналогового входа

значение: целочисленное между 0 и 255. Значение 0 генерирует 0 В на указанном порте; значение 255 генерирует +5 В на указанном порте. Для значений между 0 и 255, порт начинает быстро чередовать уровень напряжения 0 и +5 В - чем выше значение, тем, более часто порт генерирует уровень HIGH (5 В).

Пример:

```
analogWrite(9, 128); // устанавливаем на 9 контакте значение эквивалентно е 2,5В
```

Примечание:

Нет необходимости вызывать функцию `pinMode`, чтобы установить порт на вывод сигналов перед вызовом функции `analogWrite`.

Частота генерирования сигнала – приблизительно 490 Гц.

### **Работа со временем**

**millis**

Вызов:

```
time = millis();
```

Описание:

Возвращает число миллисекунд, с момента исполнения Arduino текущей программы. Счетчик переполнится и обнулится приблизительно через 9 часов.

Возвращаемое значение: возвращает значение типа unsigned long

Пример:

```
unsigned long time; // объявление переменной time типа unsigned long  
time = millis(); // передача количества миллисекунд
```

### **delay**

Вызов:

```
delay(время_мс);
```

Описание:

Приостанавливает программу на заданное число миллисекунд.

Параметры:

время\_мс – время задержки программы в миллисекундах

Пример:

```
delay(1000); //пауза 1 секунда
```

### **delayMicroseconds**

Вызов:

```
delayMicroseconds(время_мкс);
```

Описание:

Приостанавливает программу на заданное число микросекунд.

Параметры:

время\_мкс – время задержки программы в микросекундах

Пример:

```
delayMicroseconds(500); //пауза 500 микросекунд
```

### **pulseIn**

Вызов:

```
pulseIn(порт, значение);
```

Описание:

Считывает импульс (высокий или низкий) с цифрового порта и возвращает продолжительность импульса в микросекундах.

Например, если параметр «значение» при вызове функции установлен в HIGH, то pulseIn() ожидает, когда на порт поступит высокий уровень сигнала. С момента его поступления начинается отсчет времени до тех пор, пока на порт не поступит низкий уровень сигнала. Функция возвращает длину импульса (высокого уровня) в микросекундах. Работает с импульсами от 10 микросекунд до 3 минут. Обратите внимание, что эта функция не будет возвращать результат, пока импульс не будет обнаружен.

Параметры:

порт: номер порта, с которого считываем импульс

значение: тип импульса HIGH или LOW

Возвращаемое значение: возвращает длительность импульса в микросекундах (тип int)

Пример:

```
int duration; // объявление переменной duration типа int
duration = pulseIn(pin, HIGH); // измеряем длительность импульса
```

### **Последовательная передача данных**

Arduino имеет встроенный контроллер для последовательной передачи данных, который может использоваться как для связи между Arduino устройствами, так и для связи с компьютером. На компьютере соответствующее соединение представлено USB COM-портом.

Связь происходит по цифровым портам 0 и 1, и поэтому Вы не сможете использовать их для цифрового ввода/вывода если используете функции последовательной передачи данных.

#### **Serial.begin**

Вызов:

```
Serial.begin(скорость_передачи);
```

Описание:

Устанавливает скорость передачи информации COM порта битах в секунду для последовательной передачи данных. Для того чтобы поддерживать связь с компьютером, используйте одну из этих нормированных скоростей: 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, или 115200. Также Вы можете определить другие скорости при связи с другим микроконтроллером по портам 0 и 1.

Параметры:

скорость\_передачи: скорость потока данных в битах в секунду.

Пример:

```
Serial.begin(9600); //устанавливаем скорость 9600 бит/сек
```

#### **Serial.available**

Вызов:

```
count = Serial.available();
```

Описание:

Принимаемые по последовательному порту байты попадают в буфер микроконтроллера, откуда Ваша программа может их считать. Функция возвращает количество накопленных в буфере байт. Последовательный буфер может хранить до 128 байт.

Возвращаемое значение:

Возвращает значение типа int – количество байт, доступных для чтения, в последовательном буфере, или 0, если ничего не доступно.

Пример:

```
if (Serial.available() > 0) { // Если в буфере есть данные
    // здесь должен быть прием и обработка данных
}
```

#### **Serial.read**

Вызов:

```
char = Serial.read();
```

Описание:



Считывает следующий байт из буфера последовательного порта.

Возвращаемое значение:

Первый доступный байт входящих данных с последовательного порта, или -1 если нет входящих данных.

Пример:

```
incomingByte = Serial.read(); // читаем байт
```

### **Serial.flush**

Вызов:

```
Serial.flush();
```

Описание:

Очищает входной буфер последовательного порта. Находящиеся в буфере данные теряются, и дальнейшие вызовы Serial.read() или Serial.available() будут иметь смысл для данных, полученных после вызова Serial.flush().

Пример:

```
Serial.flush(); // Очищаем буфер –  
начинаем прием данных «с чистого листа»
```

### **Serial.print**

Описание:

Вывод данных на последовательный порт.

Параметры:

Функция имеет несколько форм вызова в зависимости от типа и формата выводимых данных.

Serial.print(b, DEC) выводит ASCII-строку - десятичное представление числа b.

```
int b = 79;
```

```
Serial.print(b, DEC); //выдаст в порт строку «79»
```

Serial.print(b, HEX) выводит ASCII-строку - шестнадцатеричное представление числа b.

```
int b = 79;
```

```
Serial.print(b, HEX); //выдаст в порт строку «4F»
```

Serial.print(b, OCT) выводит ASCII-строку - восьмеричное представление числа b.

```
int b = 79;
```

```
Serial.print(b, OCT); //выдаст в порт строку «117»
```

Serial.print(b, BIN) выводит ASCII-строку - двоичное представление числа b.

```
int b = 79;
```

```
Serial.print(b, BIN); //выдаст в порт строку «1001111»
```

Serial.print(b, BYTE) выводит младший байт числа b.

```
int b = 79;
```

```
Serial.print(b, BYTE); //выведет число 79 (один байт). В мониторе
```

```
//последовательного порта получим символ «О» - его
```

```
//код равен 79
```

`Serial.print(str)` если `str` – строка или массив символов, побайтно передает `str` на COM-порт.

```
char bytes[3] = {79, 80, 81}; //массив из 3 байт со значениями 79,80,81
Serial.print("Here our bytes:"); //выводит строку «Here our bytes:»
Serial.print(bytes); //выводит 3 символа с кодами 79,80,81 –
//это символы «OPQ»
```

`Serial.print(b)` если `b` имеет тип `byte` или `char`, выводит в порт само число `b`.  
`char b = 79;`

`Serial.print(b);` //выдаст в порт символ «О»

`Serial.print(b)` если `b` имеет целый тип, выводит в порт десятичное представление числа `b`.

```
int b = 79;
```

`Serial.print(b);` //выдаст в порт строку «79»

### **Serial.println**

Описание:

Функция `Serial.println` аналогична функции `Serial.print`, и имеет такие же варианты вызова. Единственное отличие заключается в том, что после данных дополнительно выводятся два символа – символ возврата каретки (ASCII 13, или `'r'`) и символ новой линии (ASCII 10, или `'n'`).

Пример 1 и пример 2 выведут в порт одно и то же:

Пример 1:

```
int b = 79;
```

```
Serial.print(b, DEC); //выдаст в порт строку «79»
```

```
Serial.print("\r\n"); //выведет символы "\r\n" – перевод строки
```

```
Serial.print(b, HEX); //выдаст в порт строку «4F»
```

```
Serial.print("\r\n");//выведет символы "\r\n" – перевод строки
```

Пример 2:

```
int b = 79;
```

```
Serial.println(b, DEC); //выдаст в порт строку «79\r\n»
```

```
Serial.println(b, HEX); //выдаст в порт строку «4F\r\n»
```

В мониторе последовательного порта получим:

79

4F

### **Материально-техническое обеспечение образовательного процесса**

Для проведения занятий используется компьютерный класс (20 компьютеров) кабинета «Информатики», при котором обязательно присутствует лаборантское помещение. В «лаборантской» осуществляется хранение оборудования Lego Mindstorms NXT (20 комплектов) и Arduino (15 комплектов). В классе имеется также мультимедийный проектор и экран, что позволяет осуществлять теоретическую подготовку обучающихся. Поскольку в классе есть отдельно стоящие ряды парт, то именно за ними обучающиеся осуществляют сборку и наладку NXT. Когда сборка робота заканчивается, они переходят за компьютеры оснащенные лицензионным программным обеспечением Lego Mindstorms (и Arduino), и производят написание и отладку

программы для работа.

Для проведения занятий достаточно наличия оборудования и программного обеспечения Lego Mindstorms (и Arduino), однако можно предложить использовать и дополнительные обучающие материалы такие, как книги по робототехнике, статьи из журналов по робототехнике, сайты по робототехнике и презентации.

## 5. ФОРМЫ АТТЕСТАЦИИ, МОНИТОРИНГ РЕЗУЛЬТАТОВ

### Мониторинг результатов обучения детей по дополнительным общеобразовательным общеразвивающим программам

Таблица № 1

Показатели (оцениваемые параметры)	Критерии	Степень выраженности оцениваемого качества	% / кол- во чел.	Методы диагностики
<b>1.Теоретическая подготовка детей:</b> 1.1. Теоретические знания (по основным разделам учебно-тематического плана программы)	Соответствие теоретических знаний программным требованиям	- <b>минимальный уровень</b> (овладели менее чем ½ объема знаний);		Соревнования, Наблюдение Итоговая работа
		- <b>средний уровень</b> (объем освоенных знаний составляет более ½);		
		- <b>максимальный уровень</b> (дети освоили практически весь объем знаний, предусмотренных программой)		
1.2. Владение специальной терминологией	Осмысленность и правильность использования	- <b>минимальный уровень</b> (избегают употреблять специальные термины);		Тестирование, Опрос, наблюдение
		- <b>средний уровень</b> (сочетают специальную терминологию с		

		бытовой);		
		- <b>максимальный уровень</b> (термины употребляют осознанно и в полном соответствии с их содержанием)		
<b>2. Практическая подготовка детей:</b> 2.1. Практические умения и навыки, предусмотренные программой (по основным разделам)	Соответствие практических умений и навыков программным требованиям	<b>минимальный уровень</b> (овладели менее чем 1/2 предусмотренных умений и навыков);		Наблюдения, Соревнования, Итоговые работы
		- <b>средний уровень</b> (объем освоенных умений и навыков составляет более 1/2);		
		- <b>максимальный уровень</b> (дети овладели практически всеми умениями и навыками, предусмотренными программой)		
2.2. Владение специальным оборудованием и оснащением	Отсутствие затруднений в использовании	- <b>минимальный уровень</b> (испытывают серьезные затруднения при работе с оборудованием)		наблюдение
		- <b>средний уровень</b> (работает с помощью педагога)		
		- <b>максимальный уровень</b> (работают самостоятельно)		
2.3. Творческие навыки	Креативность в выполнении практических заданий	- <b>начальный</b> (элементарный, выполняют лишь простейшие		Наблюдение, Итоговые работы

		практические задания)		
		- <b>репродуктивный</b> (выполняют задания на основе образца)		
		- <b>творческий</b> (выполняют практические задания с элементами творчества)		
<b>3. Общеучебные умения и навыки ребенка:</b> 3.1. Учебно-интеллектуальные умения: 3.1.1. Умение подбирать и анализировать специальную литературу	Самостоятельность в подборе и анализе литературы	<b>минимальный</b> (испытывают серьезные затруднения, нуждаются в помощи и контроле педагога) - <b>средний</b> (работают с литературой с помощью педагога и родителей) - <b>максимальный</b> (работают самостоятельно)		Наблюдение
3.1.2. Умение пользоваться компьютерными источниками информации	Самостоятельность в пользовании	Уровни по аналогии с п. 3.1.1. - <b>минимальный</b> - <b>средний</b> - <b>максимальный</b>		Наблюдение
3.1.3. Умение осуществлять учебно - исследовательскую работу (рефераты, самостоятельные учебные исследования, проекты и т.д.)	Самостоятельность в учебно-исследовательской работе	Уровни по аналогии с п. 3.1.1. - <b>минимальный</b> - <b>средний</b> - <b>максимальный</b>		Наблюдение, Инд. Работа
<b>3.2. Учебно - коммуникативные умения:</b> 3.2.1. Умение	Адекватность восприятия	Уровни по аналогии с п. 3.1.1. - <b>минимальный</b> - <b>средний</b>		Наблюдения, Опрос

слушать и слышать педагога	информации, идущей от педагога	- <b>максимальный</b>		
3.2.2. Умение выступать перед аудиторией	Свобода владения и подачи подготовленной информации	Уровни по аналогии с п. 3.1.1. - <b>минимальный</b>		наблюдения
		- <b>средний</b>		
		- <b>максимальный</b>		
<b>3.3. Учебно-организационные умения и навыки:</b> 3.3.1. Умение организовать свое рабочее (учебное) место	Самостоятельно готовят и убирают рабочее место	Уровни по аналогии с п. 3.1.1. - <b>минимальный</b>		наблюдение
		- <b>средний</b>		
		- <b>максимальный</b>		
3.3.2. Навыки соблюдения ТБ в процессе деятельности	Соответствие реальных навыков соблюдения ТБ программным требованиям	- <b>минимальный уровень</b> (овладели менее чем 1/2 объема навыков соблюдения ТБ);		наблюдение
		- <b>средний уровень</b> (объем освоенных навыков составляет более 1/2);		
		- <b>максимальный уровень</b> (освоили практически весь объем навыков)		
3.3.3. Умение аккуратно выполнять работу	Аккуратность и ответственность в работе	- <b>удовлетворительно</b>  - <b>хорошо</b>  - <b>отлично</b>		Наблюдение, Итоговые работы

**Мониторинг личностного развития детей в процессе освоения  
дополнительных общеобразовательных общеразвивающих программ**  
Таблица №2

Показатели (оцениваемые параметры)	Критерии	Степень выраженности оцениваемого качества	% / кол- во  чел.	Методы диагностики
--	----------	---	-------------------------------	-----------------------

<b>1.Организационно-волевые качества:</b> 1.1. Терпение	Способность выдерживать нагрузки, преодолевать трудности	-терпения хватает меньше чем на ½ занятия		наблюдение
		- терпения хватает больше чем на ½ занятия		
		- терпения хватает на все занятие		
1.2. Воля	Способность активно побуждать себя к практическим действиям	- волевые усилия побуждаются извне		наблюдение
		- иногда самими детьми		
		- всегда самими детьми		
1.3. Самоконтроль	Умение контролировать свои поступки	- находятся постоянно под воздействием контроля извне		наблюдение
		- периодически контролируют себя сами		
		- постоянно контролируют себя сами		
<b>2. Ориентационные качества:</b> 2.1. Самооценка	Способность оценивать себя адекватно реальным достижениям	- завышенная		Диагностические тесты
		- заниженная		
		- нормальная		
2.2. Интерес к занятиям в д/о	Осознанное участие детей в освоении образовательной программы	- интерес продиктован извне		Диагностический тест
		- интерес периодически поддерживается самим		

		- интерес постоянно поддерживается самостоятельно		
<b>3. Поведенческие качества:</b> 3.1. Конфликтность	Отношение детей к столкновению интересов (спору) в процессе взаимодействия	- периодически провоцируют конфликты		наблюдение
		- в конфликтах не участвуют, стараются их избежать		
		- пытаются самостоятельно уладить		
3.2. Тип сотрудничества (отношение детей к общим делам д/о)	Умение воспринимать общие дела, как свои собственные	- избегают участия в общих делах		Наблюдение, рабочие диагностики Н.Е.Щурковой
		- участвуют при побуждении извне		
		- инициативны в общих делах		

### Аттестация учащихся объединения «3D технологии»

**Срок проведения** промежуточной аттестация – май 2018 г.

**Цель:** оценка качества освоения учащимися программы «3D технологии» первого года обучения.

**Форма проведения:** слайд-шоу

**Содержание аттестации:**

Практическая часть аттестации – Динамическая загрузка и обновление изображений. Анимация.

**Критерии оценки промежуточного результата аттестации теория**

Низкий уровень	Средний уровень	Высокий уровень
Правильные ответы на 10 вопросов и меньше	Правильные ответы от 10 до 15 вопросов	Правильные ответы от 15 до 20 вопросов

**Критерии оценки промежуточного результата аттестации практика**

Низкий уровень	Средний уровень	Высокий уровень
Задания не выполнены	Задания выполнены с небольшими погрешностями	Задания выполнены



В том случае, если показатели по теоретической и практической части разные, то приоритет отдается результатам по практическим результатам.

### **Аттестация учащихся объединения «3D технологии»**

**Срок проведения** итоговой аттестации – май 2018 г.

**Цель:** оценка качества освоения учащимися программы «3D технологии» второго года обучения.

**Форма проведения:** проект на РНР

**Содержание аттестации:**

Практическая часть аттестации – проект на РНР

Теоретическая часть - Тестовые задания по всем основным темам за прошедший год

**Критерии оценки итогового результата аттестации теория**

<b>Низкий уровень</b>	<b>Средний уровень</b>	<b>Высокий уровень</b>
Правильные ответы на 10 вопросов и меньше	Правильные ответы от 10 до 15 вопросов	Правильные ответы от 15 до 20 вопросов

**Критерии оценки итогового результата аттестации практика  
(квалификационные заезды)**

<b>Низкий уровень</b>	<b>Средний уровень</b>	<b>Высокий уровень</b>
Задания не выполнены	Задания выполнены с небольшими погрешностями	Задания выполнены

В том случае, если показатели по теоретической и практической части разные, то приоритет отдается результатам по практической части.

## 6. СПИСОК ЛИТЕРАТУРЫ

Программа составлена с использованием материала

1. Сайта Robocraft.ru
2. Программного обеспечения компании Lego и сообщества Arduino в России.

### А) Литература для педагога

#### *основная*

1. Глинский Б. А. Моделирование как метод научного исследования. — М.: 1965.
2. Техническое творчество. Программы для внешкольных учреждений и общеобразовательных школ. — М.: Просвещение, 1978.
3. Программа образовательной области «Технология». — М.: ВНК «Технология», 1996
4. Артоболевский И. И. Механизмы в современной технике. — М.: Наука, 1970.

#### *дополнительная*

1. Ханзен Р. Основы общей методики конструирования. — М.: Знание, 1968.
2. Бессонов В. Кружок радиоэлектроники. — М.: Просвещение, 1993-
3. Борисов В. Кружок радиотехнического конструирования. — М.: Радио и связь, 1989.
4. Варламов Р. Мастерская радиолюбителя. — М.: Радио и связь, 1983.
5. Иванов Б. Энциклопедия начинающего радиолюбителя, — М., 1992.
6. Программы для внешкольных учреждений. Технические кружки по электронике, микропроцессорной технике. — М.: Просвещение, 1987.
7. Фролов В. Язык радиосхем. — М.: Радио и связь, 1989.
8. Эндерлайн Р. Микроэлектроника для всех. — М: Мир, 1989

### Б) Литература для обучающихся

#### *Основная*

1. Блог А. Колотова "NiNoNXT"
2. Вводный курс по программированию NXT. Часть I. Учебное пособие научно-технической конференции LEGO "Инженерная культура: от школы к производству" Научно-Методический Центр Университета TUFTS, 2012. 40 с.
3. Вводный курс по программированию NXT. Часть II. Учебное пособие научно-технической конференции LEGO "Инженерная культура: от школы к производству" Научно-Методический Центр Университета TUFTS, 2012. 34 с.
4. Копосов Д. Г. Первый шаг в робототехнику: практикум для 5-6 классов М.: БИНОМ. Лаборатория знаний, 2012. 286 с.
5. Сервомотор [Электронный ресурс] Lego education: [сайт]

6. Филиппов С. А. Робототехника для детей и родителей СПб.: Наука, 2013. 319 с.
7. NXT ® motor internals [Электронный ресурс] Philo's Home Page: [сайт]  
*дополнительная*
1. Вильямс Д. Программируемый робот, управляемый с КПК /Д. Вильямс; пер. с англ. А. Ю. Карцева. — М.: НТ Пресс, 2006. — 224 с; ил. (Робот — своими руками).
2. Комский Д. Кружок технической кибернетики. — М.: Просвещение, 1991.
3. Мацкевич. Занимательная анатомия роботов. — 2-е изд., перераб. и доп. — М.: Радио и связь», 1988. — 128 с; ил. — (Межизд. серия «Научно-популярная библиотека школьника»).
4. Хейзерман Д. Как самому сделать робота: Пер. с англ. В. С. Гурфинкеля. — М.: Мир, 1979.

## 7. Календарный учебный график

## 1-й год обучения

№ п/п	Месяц	Число	Время проведения занятия	Форма занятия	Кол-во часов	Тема занятия	Место проведения	Форма контроля
1.				Лекция	2	Введение в робототехнику. История робототехники. Конструкторы компании ЛЕГО. Знакомимся с набором Lego Mindstorms NXT 2.0 сборки 8547	Центр робототехники	Устный опрос
2.				Практическое занятие	2	Калибровка датчика. Шины данных. Шины данных - Повреждение	Центр робототехники	Устный опрос
3.				Лекция	2	Мультизагрузка NXT. Файлы и память устройства NXT	Центр робототехники	Устный опрос
4.				Практическое занятие	2	Профили. Цепочка программы	Центр робототехники	Устный опрос
5.				Практическое занятие	2	Простой текст. Горячие клавиши NXT	Центр робототехники	Устный опрос
6.				Лекция	2	Начальная точка. Обновление операционной системы NXT	Центр робототехники	Устный опрос
7.				Лекция	2	Создание Проекта	Центр робототехники	Устный опрос
8.				Практическое занятие	2	Блок «Экран».	Центр робототехники	Устный опрос
9.				Практическое занятие	2	Блок «Движение». Блок «Писать/Играть»	Центр робототехники	Устный опрос
10.				Практическое занятие	2	Блок «Переключатель»	Центр робототехники	Устный опрос
11.				Практическое занятие	2	Блок «Ожидание».	Центр робототехники	Устный опрос
12.				Практическое	2	Блок «Лампа». Блок «Мотор»	Центр	Устный опрос

				ое занятие			робототехники	
13.				Практическое занятие	2	Блок «Отправить сообщение»	Центр робототехники	Устный опрос
14.				Практическое занятие	2	Блок «Приём сообщений»	Центр робототехники	Устный опрос
15.				Практическое занятие	2	Блок «Звук»	Центр робототехники	Устный опрос
16.				Практическое занятие	2	Блок «Датчик освещённости»	Центр робототехники	Устный опрос
17.				Практическое занятие	2	Блок «Кнопки NXT»	Центр робототехники	Устный опрос
18.				Практическое занятие	2	Блок «Датчик оборотов». Блок «Датчик звука»	Центр робототехники	Устный опрос
19.				Практическое занятие	2	Блок «Датчик температуры».	Центр робототехники	Устный опрос
20.				Практическое занятие	2	Блок «Таймер»	Центр робототехники	Устный опрос
21.				Практическое занятие	2	Блок «Датчик касания»	Центр робототехники	Устный опрос
22.				Практическое занятие	2	Блок «Датчик расстояния»	Центр робототехники	Устный опрос
23.				Практическое занятие	2	Блок «Цикл»	Центр робототехники	Устный опрос
24.				Практическое занятие	2	Блок «Стоп».	Центр робототехники	Устный опрос
25.				Практическое занятие	2	Блок «Сравнение»	Центр робототехники	Устный опрос
26.				Практическое занятие	2	Блок «Логика». Блок «Математика»	Центр робототехники	Устный опрос
27.				Практическое занятие	2	Блок «Случайное число». Блок «Интервал»	Центр робототехники	Устный опрос
28.				Практическое занятие	2	Блок «Переменная». Блок «Константа»	Центр робототехники	Устный опрос
29.				Практическое	2	Блок «Калибровка». Блок «Доступ к	Центр	Устный опрос

				ое занятие		файлам»	робототехники	
30.				Практическое занятие	2	Блок «Не засыпать». Блок «Число в текст»	Центр робототехники	Устный опрос
31.				Практическое занятие	2	Блок «Обнулить мотор». Блок «Текст»	Центр робототехники	Устный опрос
32.				Практическое занятие	2	Блок «Bluetooth соединение».	Центр робототехники	Устный опрос
33.				Практическое занятие	2	Блок «Начать регистрацию данных» Блок «Остановить регистрацию данных». Анализ данных	Центр робототехники	Устный опрос
34.				Практическое занятие	2	Панель параметров эксперимента. Пульт регистрации данных	Центр робототехники	Устный опрос
35.				Практическое занятие	2	Таблица данных. График	Центр робототехники	Устный опрос
36.				Практическое занятие	2	Управление файлами экспериментов. Создание прогноза. Удаление данных	Центр робототехники	Устный опрос
37.				Лекция	2	Программа. Сборка. Класс	Центр робототехники	Устный опрос
38.				Практическое занятие	2	Система типов	Центр робототехники	Устный опрос
39.				Лекция	2	Управляющие операторы и методы	Центр робототехники	Устный опрос
40.				Практическое занятие	2	Объекты	Центр робототехники	Устный опрос
41.				Практическое занятие	2	Массивы	Центр робототехники	Устный опрос
42.				Лекция	2	Перегруженные операции	Центр робототехники	Устный опрос
43.				Лекция	2	Наследование и полиморфизм	Центр робототехники	Устный опрос
44.				Практическое занятие	2	Интерфейсы	Центр робототехники	Устный опрос
45.				Практическое занятие	2	Делегаты и события	Центр робототехники	Устный опрос

46.				Практическое занятие	2	Атрибуты, сборки, рефлексия	Центр робототехники	Устный опрос
47.				Практическое занятие	2	Ввод/вывод	Центр робототехники	Устный опрос
48.				Практическое занятие	2	Коллекции. Параметризованные классы	Центр робототехники	Устный опрос
49.				Практическое занятие	2	Шаблоны	Центр робототехники	Устный опрос
50.				Практическое занятие	2	Совмещение управляемого и неуправляемого кодов	Центр робототехники	Устный опрос
51.				Практическое занятие	2	Потоки	Центр робототехники	Устный опрос
52.				Практическое занятие	2	Форма	Центр робототехники	Устный опрос
53.				Практическое занятие	2	GDI+	Центр робототехники	Устный опрос
54.				Практическое занятие	2	Основы ADO .NET	Центр робототехники	Устный опрос
55.				Практическое занятие	2	Основные понятия	Центр робототехники	Устный опрос
56.				Практическое занятие	2	Обработчики событий: onMouseover	Центр робототехники	Устный опрос
57.				Практическое занятие	2	Запросы пользователю и переменные	Центр робототехники	Устный опрос
58.				Практическое занятие	2	Концепция свойств	Центр робототехники	Устный опрос
59.				Практическое занятие	2	Создание функций	Центр робототехники	Устный опрос
60.				Практическое занятие	2	Команды последствия: onUnload и onMouseOut	Центр робототехники	Устный опрос
61.				Практическое занятие	2	Открываем новые окна	Центр робототехники	Устный опрос
62.				Практическое занятие	2	Открытие окна с помощью функции	Центр робототехники	Устный опрос

63.				Практическое занятие	2	Метод 'Confirm' (Введение в if и else)	Центр робототехники	Устный опрос
64.				Практическое занятие	2	Математические вычисления	Центр робототехники	Устный опрос
65.				Практическое занятие	2	Изменение изображения с помощью события onMouseOver	Центр робототехники	Устный опрос
66.				Практическое занятие	2	Вызов функции в формы	Центр робототехники	Устный опрос
67.				Практическое занятие	2	Передача данных в функцию	Центр робототехники	Устный опрос
68.				Практическое занятие	2	Оператор if и ветвление	Центр робототехники	Устный опрос
69.				Практическое занятие	2	Случайный выбор фраз и изображений	Центр робототехники	Устный опрос
70.				Практическое занятие	2	Введение в циклы	Центр робототехники	Устный опрос
71.				Практическое занятие	2	Массивы	Центр робототехники	Устный опрос
72.				Практическое занятие	2	Соревнование	Центр робототехники	Саморефлексия

## 2-й год обучения

№ п/п	Месяц	Число	Время проведения занятия	Форма занятия	Кол-во часов	Тема занятия	Место проведения	Форма контроля
1.				Лекция	2	Программирование Arduino - введение	Центр робототехники	Устный опрос
2.				Практическое занятие	2	Структура программы, константы	Центр робототехники	Устный опрос
3.				Лекция	2	Цифровой ввод/вывод	Центр робототехники	Устный опрос



4.				Практическое занятие	2	Аналоговый ввод/вывод	Центр робототехники	Устный опрос
5.				Практическое занятие	2	Дополнительные функции ввода/вывода	Центр робототехники	Устный опрос
6.				Лекция	2	Работа со временем	Центр робототехники	Устный опрос
7.				Лекция	2	Математические функции	Центр робототехники	Устный опрос
8.				Практическое занятие	2	Псевдослучайные числа	Центр робототехники	Устный опрос
9.				Практическое занятие	2	Последовательная передача данных	Центр робототехники	Устный опрос
10.				Практическое занятие	2	Прерывания	Центр робототехники	Устный опрос
11.				Практическое занятие	2	EEPROM	Центр робототехники	Устный опрос
12.				Практическое занятие	2	Создание своей библиотеки	Центр робототехники	Устный опрос
13.				Практическое занятие	2	Blink без delay	Центр робототехники	Устный опрос
14.				Практическое занятие	2	Создание библиотеки. Ethernet library – библиотека для работы с Ethernet	Центр робототехники	Устный опрос
15.				Практическое занятие	2	Создание библиотеки. Servo library – библиотека для работы с сервомашинками	Центр робототехники	Устный опрос
16.				Практическое занятие	2	Создание библиотеки. Firmata library – библиотека, реализующая протокол Firmata	Центр робототехники	Устный опрос
17.				Практическое занятие	2	Создание библиотеки. OneWireSlave - библиотека эмуляции устройства 1-Wire	Центр робототехники	Устный опрос
18.				Практическое занятие	2	Arduino и PROTEUS. Arduino и WinAVR - программируем на чистом C	Центр робототехники	Устный опрос
19.				Практическое занятие	2	Цифровой ввод - кнопка	Центр робототехники	Устный опрос

20.				Практическое занятие	2	Аналоговый вывод - Fading	Центр робототехники	Устный опрос
21.				Практическое занятие	2	Аналоговый ввод – потенциометр	Центр робототехники	Устный опрос
22.				Практическое занятие	2	Аналоговый ввод – осциллограф	Центр робототехники	Устный опрос
23.				Практическое занятие	2	Генерация звука – пьезоизлучатель.	Центр робототехники	Устный опрос
24.				Практическое занятие	2	Фоторезистор	Центр робототехники	Устный опрос
25.				Практическое занятие	2	Сенсор на светодиоде	Центр робототехники	Устный опрос
26.				Практическое занятие	2	Общение с Arduino - программирование работы с COM-портом.	Центр робототехники	Устный опрос
27.				Практическое занятие	2	Аналоговый датчик температуры – LM335	Центр робототехники	Устный опрос
28.				Практическое занятие	2	Подключаем к Arduino мышку PS/2	Центр робототехники	Устный опрос
29.				Практическое занятие	2	Протокол 1-Wire и iButton	Центр робототехники	Устный опрос
30.				Практическое занятие	2	Arduino и эмулятор iButton	Центр робототехники	Устный опрос
31.				Практическое занятие	2	Arduino и температурный 1-Wire датчик DS18S202	Центр робототехники	Устный опрос
32.				Практическое занятие	2	Arduino и драйвер двигателей L293D (Простой мотор-шилд)	Центр робототехники	Устный опрос
33.				Практическое занятие	2	Arduino и сервомашинка	Центр робототехники	Устный опрос
34.				Практическое занятие	2	Как с помощью Arduino можно управлять устройствами на 220В	Центр робототехники	Устный опрос
35.				Практическое занятие	2	Подключаем LCD-дисплей на базе HD44780 к Arduino	Центр робототехники	Устный опрос
36.				Практическое занятие	2	ИК-датчик препятствий для Arduino на базе фототранзистора	Центр робототехники	Устный опрос

37.				Лекция	2	Что такое ASP .NET	Центр робототехники	Устный опрос
38.				Практическое занятие	2	Анатомия ASP .NET. ASP .NET в действии	Центр робототехники	Устный опрос
39.				Лекция	2	Серверные элементы управления	Центр робототехники	Устный опрос
40.				Практическое занятие	2	Серверные элементы управления (продолжение)	Центр робототехники	Устный опрос
41.				Практическое занятие	2	AutoPostBack. Привязка к данным. Коллекции. Проверка правильности вводимых данных	Центр робототехники	Устный опрос
42.				Лекция	2	Работа с базами данных	Центр робототехники	Устный опрос
43.				Лекция	2	Работа с базами данных (продолжение). Элементы-источники данных (Data Source Controls)	Центр робототехники	Устный опрос
44.				Практическое занятие	2	Элементы-потребители данных. Data-Bound Controls	Центр робототехники	Устный опрос
45.				Практическое занятие	2	Элементы-потребители данных ASP .NET 2.0	Центр робототехники	Устный опрос
46.				Практическое занятие	2	Работа с XML	Центр робототехники	Устный опрос
47.				Практическое занятие	2	Навигация по сайту	Центр робототехники	Устный опрос
48.				Практическое занятие	2	Шаблоны дизайна страниц ASP .NET	Центр робототехники	Устный опрос
49.				Практическое занятие	2	Стили элементов управления. Темы и шкурки	Центр робототехники	Устный опрос
50.				Практическое занятие	2	Пользовательские и собственные серверные элементы управления	Центр робототехники	Устный опрос
51.				Практическое занятие	2	Конфигурирование проекта. Глобальные объекты	Центр робототехники	Устный опрос
52.				Практическое занятие	2	Web-сервисы	Центр робототехники	Устный опрос

53.				Практическое занятие	2	Локализация проектов. Аутентификация пользователей	Центр робототехники	Устный опрос
54.				Практическое занятие	2	Элементы управления и использование JavaScript	Центр робототехники	Устный опрос
55.				Практическое занятие	2	Введение в PHP	Центр робототехники	Устный опрос
56.				Практическое занятие	2	Области применения и способы использования. Создание первого скрипта на PHP. Основы синтаксиса	Центр робототехники	Устный опрос
57.				Практическое занятие	2	Управляющие конструкции	Центр робототехники	Устный опрос
58.				Практическое занятие	2	Обработка запросов с помощью PHP	Центр робототехники	Устный опрос
59.				Практическое занятие	2	Характеристика методов Post и Get.	Центр робототехники	Устный опрос
60.				Практическое занятие	2	Функции в PHP	Центр робототехники	Устный опрос
61.				Практическое занятие	2	Объекты и классы в PHP	Центр робототехники	Устный опрос
62.				Практическое занятие	2	Работа с массивами данных	Центр робототехники	Устный опрос
63.				Практическое занятие	2	Работа со строками	Центр робототехники	Устный опрос
64.				Практическое занятие	2	Работа с файловой системой	Центр робототехники	Устный опрос
65.				Практическое занятие	2	Базы данных и СУБД. Введение в SQL	Центр робототехники	Устный опрос
66.				Практическое занятие	2	Взаимодействие PHP и MySQL	Центр робототехники	Устный опрос
67.				Практическое занятие	2	Авторизация доступа с помощью сессий	Центр робототехники	Устный опрос
68.				Практическое занятие	2	Настройка сессий в файлах php.ini, httpd.conf, .htaccess.	Центр робототехники	Устный опрос
69.				Практическое	2	Регулярные выражения	Центр	Устный опрос

				ое занятие			робототехники	
<b>70.</b>				Практическ ое занятие	<b>2</b>	Взаимодействие PHP и XML	Центр робототехники	Устный опрос
<b>71.</b>				Практическ ое занятие	<b>2</b>	Использование шаблонов в PHP	Центр робототехники	Устный опрос
<b>72.</b>				Практическ ое занятие	<b>2</b>	Проект на PHP.	Центр робототехники	Саморефлексия

