

МИНИСТЕРСТВО ОБРАЗОВАНИЯ КУЗБАССА  
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО  
ОБРАЗОВАНИЯ «КУЗБАССКИЙ ЦЕНТР ДОМ ЮНАРМИИ»

Принята на заседании  
педагогического совета  
«30» августа 2023 г.  
Протокол № 1



Дополнительная общеобразовательная общеразвивающая программа  
технической направленности

**«РОБОТОТЕХНИКА И ПРОГРАММИРОВАНИЕ»**

для обучающихся 9–17 лет

Срок реализации - 2 года

Уровень программы – базовый



Разработчик: Макаренко Данила  
Александрович, педагог  
дополнительного образования

г. Кемерово, 2023 г.

## Содержание

Комплекс основных характеристик программы.....	3
Пояснительная записка .....	3
Учебно-тематический план 1 год обучения.....	6
Содержание учебно-тематического плана 1 год обучения .....	9
Учебно-тематический план 2 год обучения.....	13
Содержание учебно-тематического плана 2 год обучения .....	15
Планируемые результаты .....	21
Комплекс организационно-педагогических условий.....	23
Календарный учебный график .....	23
Условия реализации программы .....	24
Методические материалы .....	26
Список использованной литературы и источников: .....	28

## **Комплекс основных характеристик программы**

### **Пояснительная записка**

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Робототехника и программирование» (далее - Программа) предусматривает базовый уровень освоения содержания, позволяющий обучающимся приобрести базовый минимум знаний, умений и навыков по робототехнике и программированию

Данная программа разработана в соответствии с нормативно-правовыми документами:

– Федеральный Закон от 29.12.2012 № 273-ФЗ «Об образовании в РФ».

– Концепция развития дополнительного образования детей до 2030 года (Распоряжение Правительства Российской Федерации от 31 марта 2022 г. № 678-р)

– Постановление Главного государственного санитарного врача РФ от 28 сентября 2020 года N 28 Об утверждении санитарных правил СП 2.4.3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи"

– Государственная программа РФ «Развитие образования» на 2018 - 2025 гг. (постановление Правительства Российской Федерации от 26 декабря 2017 г. № 1642);

– Профессиональный стандарт «Педагог дополнительного образования детей и взрослых» (Приказ Минтруда и соц. защиты РФ от 05.05.2018 № 298н);

– Приказ Министерства просвещения Российской Федерации от 27 июля 2022 года № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

– Письмо Министерства образования и науки РФ от 18.11.2015 № 09-3242 «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы).

– «Целевая модель развития региональных систем дополнительного образования детей» (утверждена приказом Министерства просвещения РФ № 467 от 3 сентября 2019 года).

– Устав и локальные нормативные акты образовательной организации, регламентирующие учебный процесс.

#### **Актуальность программы.**

В условиях невысокой мотивации детей к познанию и научно-техническому творчеству, низкому престижу инженерных специальностей

особую актуальность приобретает совершенствование дополнительных образовательных программ, создание программ для особого развивающего пространства и форм для интеллектуального развития детей и молодежи, их подготовка по программам инженерной направленности.

Мотивацию детей к научно-техническому творчеству можно развивать при помощи образовательной робототехники, т. к. робототехника на сегодняшний момент является одним из направлений, способных объединить в себе фактически все школьные предметы естественнонаучного цикла, реализовать и укрепить межпредметные связи.

**Отличительной особенностью** программы является ее модульность и практическая направленность. Обучающиеся по программе учатся основам механики, алгоритмизации, построению блоксхем, программированию микроконтроллеров. Все практические занятия, включенные в модули программы проводятся на реальных конструкторах серии LEGO Mindstorms, с помощью которых обучающиеся учатся построению роботизированных манипуляторов и самоходных автоматов, выполняющих заданные функции.

Содержание программных модулей способствует развитию системы универсальных учебных действий в составе личностных, регулятивных, познавательных и коммуникативных действий. Особое внимание уделяется математическим исследованиям и построению алгоритмов. Важный компонент занятий - практическое применение сконструированных моделей

**Педагогическая целесообразность программы** основывается на идее педагогической поддержки обучающихся в творческом развитии и реализации их творческого потенциала. Образовательные модули предназначены для изучения основ робототехники, организации проектной деятельности, моделирования и технического творчества обучающихся; способствуют освоению базовых навыков в области проектирования и моделирования объектов; направлены на стимулирование и развитие любознательности и интереса к технике.

По программе предполагается логичное соблюдение принципов, позволяющих учитывать разный уровень развития и разную степень освоения обучающимися содержания. Каждый программный модуль самостоятелен, может быть освоен обучающимися как отдельная составляющая с формализованными конкретными результатами обучения и формами контроля.

При комплексном освоении программных модулей осуществляется целостное освоение содержания, при котором достигается основная цель программы.

При разработке данной модульной программы учтены принципы, позволяющие учитывать разный уровень развития и разную степень освоения программного содержания обучающимися.

Программа опирается на систему педагогических принципов:

- принцип развивающего и воспитывающего обучения;
- принцип деятельностного подхода;
- принцип добровольности;
- принцип сознательности и активности обучающихся;
- принцип дифференцированного обучения;
- принцип наглядности и доступности;
- принцип «от простого к сложному»;
- принцип систематичности и последовательности;
- принцип связи теории с практикой;
- принцип социокультурного соответствия;
- принцип психологической комфортности в коллективе.

#### **Адресат программы**

Программа направлена на привлечение обучающихся к современным технологиям конструирования, программирования и использования роботизированных устройств. Возраст детей, участвующих в реализации программы от 9 до 17 лет. В коллектив могут быть приняты все желающие осваивать данное направление. Группы 2-го года обучения комплектуется, как правило, из обучающихся, прошедших 1 год. Состав группы 8 человек, определяется количеством рабочих мест, сложностью применяемого оборудования, нормами санитарного законодательства.

**Объем программы 432 часа, срок освоения 2 учебных года**

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

**Форма обучения** очная, занятия, групповые и индивидуальные определяются содержанием Программы и могут предусматривать практические занятия, мастер-классы, выполнение самостоятельной работы, выставки, творческие отчеты, соревнования и другие виды учебных занятий

**Цель программы:** обучение основам робототехники и программирования обучающихся, мотивированных в дальнейшем на профессиональную подготовку, связанную с данным направлением, развитие творческих (инженерных, технических) способностей в процессе конструирования и проектирования,

#### **Задачи**

**Образовательные:**

- дать первоначальные знания по устройству робототехнических устройств;

- научить основным приемам сборки и программирования робототехнических средств;
- сформировать общенаучные и технологические навыки конструирования и проектирования;
- ознакомить с правилами безопасной работы с инструментами необходимыми при конструировании робототехнических средств.

***Развивающие:***

- развивать творческую инициативу и самостоятельность;
- развивать психофизиологические качества обучающихся: память, внимание, способность логически мыслить, анализировать, концентрировать внимание на главном.

***Воспитательные:***

- формировать творческое отношение по выполняемой работе;
- воспитывать умение работать в коллективе;
- формировать у обучающихся стремления к получению качественного законченного результата;
- формировать навыки проектного мышления.

### Учебно-тематический план 1 год обучения

№ п/п	Название темы	Количество часов:			Формы контроля
		Теория	Практика	Всего	
1	<b>Введение в робототехнику и программирование</b>	3	-	3	---
2	<b>Конструктор Lego Mindstorms NXT 2.0 – 9797</b>	9	36	45	<b>Решение задач программирования.</b>
2.1	Правила работы с конструктором LEGO Mindstorms NXT 2.0.	3	0	3	Опрос
2.2	Основные детали.	3	0	3	Тестирование
2.3	Спецификация. Кнопки управления.	1	8	9	Опрос
2.4	Назначение портов NXT 2.0.	1	5	6	Опрос

2.5	Знакомство с датчиками.	1	11	12	Опрос
2.6	Программирование во внутренней среде робота	3	9	12	Решение задач программирования
<b>3</b>	<b>Введение в механику</b>	<b>18</b>	<b>18</b>	<b>36</b>	<b>Упражнения, решение механических задач, проекты</b>
3.1	История механики, основные понятия и термины.	0	3	3	Упражнения, решение механических задач
3.2	Робот пятиминутка	0	3	3	Упражнения, решение механических задач
3.3	Робот внедорожника	0	3	3	Упражнения, решение механических задач
3.4	Робот исследователь	0	3	3	Упражнения, решение механических задач
3.5	Робот автобот	0	3	3	Упражнения, решение механических задач
3.5	Свой робот	3	18	21	Упражнения, проект
<b>4</b>	<b>Введение в NXT-G</b>	<b>15</b>	<b>21</b>	<b>36</b>	<b>Решение задач на программирование. Соревнования.</b>
4.1	Окно инструментов, Команды NXT-G	3	6	9	Решение задач на программирование.
4.2	Работа с пиктограммами, Соединение команд	3	6	9	Решение задач на программирование.

4.3	Составление линейных программ, передача и запуск программы.	3	6	9	Решение задач на программирование. Соревнования
4.4	Составление программы с использованием параметров, программы с циклом. Условие, условный переход. Датчики и их параметры	3	6	9	Решение задач на программирование. Соревнования
<b>5</b>	<b>Блок движения</b>	<b>9</b>	<b>36</b>	<b>45</b>	<b>Выполнение упражнений, решение задач. Соревнования.</b>
5.1	Использование градусов, оборотов, секунд и «без ограничения» в блоке движения	3	12	15	Выполнение упражнений, решение задач. Соревнования.
5.2	Программирование автономного робота на объезд препятствий	3	12	15	Выполнение упражнений, решение задач. Соревнования.
5.3	Выполнение блока движения при определённых условиях	0	15	15	Выполнение упражнений, решение задач. Соревнования.
<b>6</b>	<b>Блок ожидания</b>	<b>3</b>	<b>12</b>	<b>15</b>	<b>Выполнение упражнений, решение задач. Соревнования.</b>
<b>7</b>	<b>Блок цикл</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>Выполнение упражнения, решение задачи.</b>
<b>8</b>	<b>Блок переключатель</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>Выполнение упражнения, решение задачи</b>
<b>9</b>	<b>Блок звука</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>Выполнение упражнения, решение задачи</b>



10	Блок дисплея	1	2	3	Выполнение упражнения, решение задачи
11	Блок записи/воспроизведения	1	2	3	Выполнение упражнения, решение задачи
12	Проект на заданную тему	1	11	12	Защита проекта
13	Заключительное занятие	3	-	3	---
14	Резерв	-	6	-	---
	<b>ИТОГО часов:</b>	<b>66</b>	<b>150</b>	<b>216</b>	<b>---</b>

## Содержание учебно-тематического плана

### 1-го года обучения

#### 1. Введение в робототехнику и программирование.

**Теория:** История робототехники и их программирования. Что такое робот. Робот и современность. Виды языков программирования. Виды роботов. Перспективы развития робототехники и программирования.

**Практика:** Запись

#### 2. Конструктор Lego Mindstorms NXT 2.0 – 9797.

##### 2.1 Правила работы с конструктором Lego Mindstorms NXT 2.0 – 9797.

**Теория:** Знакомство с набором Lego Mindstorms NXT 2.0 – 9797, техника безопасности и правила при использовании конструктора Lego Mindstorms NXT 2.0 – 9797.

**Практика:** Запись правил.

##### 2.2 Основные детали.

**Теория:** Изучение таких деталей как, балка, ось, пластина, штифт, захват, втулка, фиксатор, рукоятка и т.д.

**Практика:** Запись правил.

##### 2.3 Спецификация кнопки управления.

**Теория:** Кнопки управления NXT Оранжевая: включить/Ввести. Светло-серые стрелки: Навигация, влево и Вправо. Тёмно-серая: очистить/Вернуться.

**Практика:** Опрос.

## **2.4 Назначение портов NXT 2.0.**

**Теория:** Выходные порты робота, их устройство и назначение. Пометки, USB – порт и его назначение. Способ подключения робота к компьютеру без портов с использованием Bluetooth.

**Практика:** Выполнение практических задач.

## **2.5 Знакомство с датчиками.**

**Теория:** Изучение датчиков касания, звука, освещённости, расстояния, интерактивных сервомоторов.

**Практика:** Использование каждого датчика.

## **2.6 Программирование во внутренней среде.**

**Теория:** Изучение блоков движения, касания, света, темноты, звука, времени. Назначение и пример работы.

**Практика:** Выполнение практических задач.

## **3 Введение в механику.**

### **3.1 История механики, основные понятия и термины.**

**Теория:** История зарождения первых механизмов, первые и величайшие механики в истории человечества. Современная механика, самый сложный и самый простой механизм в мире.

**Практика:** Запись

### **3.2 Робот пятиминутка.**

**Теория:** Изучение деталей для конструирования.

**Практика:** Конструирование робота пятиминутки.

### **3.3 Робот внедорожник.**

**Теория:** Изучение деталей для конструирования.

**Практика:** Конструирование робота внедорожника.

### **3.4 Робот исследователь.**

**Теория:** Изучение деталей для конструирования.

**Практика:** Конструирование робота исследователя.

### **3.5 Робот автобот.**

**Теория:** Изучение деталей для конструирования.

**Практика:** Конструирование робота автобота.

### **3.6 Свой робот.**

**Теория:** Подбор деталей для своего робота, составление плана, придумывание названия.

**Практика:** Конструирование своего робота.

## **4 Введение в NXT-G.**

### **4.1 Окно инструментов, команды NXT-G.**

**Теория:** Изучение интерфейса программы, правила создания своего собственного профиля. Инструменты.

**Практика:** Выполнение практических задач.

### **4.2 Работа с пиктограммами, соединение команд.**

**Теория:** Изучение пиктограмм, настройки каждой, пример использования и исполнения соединив перед этим.

**Практика:** Опрос.

### **4.3 Составление линейных программ, передача и запуск программы.**

**Теория:** Примеры линейных программ. Передача программы через USB порт, подключенный к роботу. Рассказ как запустить программу через компьютер или внутри робота.

**Практика:** Выполнение практических задач.

### **4.4 Составление программы с использованием параметров, программы с циклом. Условия, условный переход, датчики и их параметры.**

**Теория:**

**Практика:**

## **5. Блок движения.**

### **5.1 Использование градусов, оборотов, секунд и «без ограничения» в блоке движения.**

**Теория:** Разница между единицами измерения прокрутки колёс, показ на примере запуска робота.

**Практика:**

## **5.2 Программирование автономного робота на объезд препятствий.**

**Теория:** Составление плана, показ препятствия, обнаружение трудностей и проблем перед началом составления программы.

**Практика:** Запуск робота.

## **5.3 Выполнение блока движения при определённых условиях.**

**Теория:** Изучения блока движения при условии света, темноты, громкого звука, касания, расстояния до объекта.

**Практика:** Выполнение практических задач.

## **6 Блок ожидания.**

**Теория:** Понятие блока ожидания, пример работы, его значение и назначение в робототехнике и программировании.

**Практика:** Выполнение практических задач.

## **7 Блок цикл.**

**Теория:** Понятие блока цикла, пример работы, его значение и назначение в робототехнике и программировании.

**Практика:** Выполнение практических задач.

## **8 Блок переключатель.**

**Теория:** Понятие блока переключателя, его настройки, пример работы, его значение и назначение в робототехнике и программировании.

**Практика:** Выполнение практических задач.

## **9 Блок звука.**

**Теория:** Понятие блока звука, его настройки, пример работы, его значение и назначение в робототехнике и программировании.

**Практика:** Выполнение практических задач.

## **10 Блок дисплея.**

**Теория:** Понятие блока дисплея, его настройки, пример работы, его значение и назначение в робототехнике и программировании.

**Практика:** Выполнение практических задач.

### **11 Блок записи/воспроизведения.**

**Теория:** Понятие блока цикла, пример работы, его значение и назначение в робототехнике и программировании.

**Практика:** Выполнение практических задач.

### **12 Проект на заданную тему.**

**Теория:** Постановка задачи проекта условия выполнения, ограничения. Выяснение всех непонятных вопросов.

**Практика:** Выполнение проекта.

### **13 Заключительное занятие: Подведение итогов работы.**

## **Учебно-тематический план 2 год обучения**

№ п/п	Наименование разделов и тем	Количество часов			Формы контроля
		всего	теория	практика	
<b>1</b>	<b>Вводное занятие</b>	<b>3</b>	3	<b>0</b>	устный опрос
<b>2</b>	<b>Блок – схемы</b>	<b>12</b>	<b>4</b>	<b>8</b>	опрос, практическая работа
2.1	Функции блоков	3	3		
2.2	Линейный алгоритм	3		3	
2.3	Разветвляющийся алгоритм	3		3	
2.4	Циклический алгоритм	3		3	
<b>3</b>	<b>Основы программирования</b>	<b>9</b>	3	<b>6</b>	
3.1	Переменные и их виды	9	3	6	опрос, практическая работа
<b>4</b>	<b>Операторы и выражения</b>	<b>6</b>	<b>2</b>	<b>4</b>	опрос, практическая работа
4.1	Арифметические операторы	1		1	
4.2	Операторы сравнения	1		1	

4.3	Операторы присваивания	1		1	
4.4	Логические операторы	1		1	
4.5	Операторы принадлежности	1		1	
4.6	Операторы тождественности	1		1	
<b>5</b>	<b>Поток команд</b>	<b>15</b>	<b>6</b>		опрос, практическая работа
5.2	Цикл while	6	3	3	
5.3	Цикл for	9	3	6	
<b>6</b>	<b>Функции</b>	<b>18</b>	<b>6</b>	<b>12</b>	
6.1	Аргументы функции	12	6	6	опрос, практическая работа
6.2	Анонимные функции, инструкция lambda	6	3	3	опрос, практическая работа
<b>7</b>	<b>Модули</b>	<b>21</b>	<b>6</b>	<b>15</b>	опрос, практическая работа
7.1	Модуль csv – чтение и запись файлов	12	3	9	
7.2	Модуль unittest: тестируем свои программы	9	3	6	
<b>8</b>	<b>Структуры данных</b>	<b>30</b>	<b>10</b>	<b>20</b>	
8.1	Список, кортеж, словарь	12	6	6	
8.2	Последовательности, множества, ссылки	18	4	14	
<b>9</b>	<b>Решение задач</b>	<b>16</b>	<b>4</b>	<b>12</b>	
<b>10</b>	<b>Объектно – ориентированное программирование</b>	<b>30</b>	<b>10</b>	<b>20</b>	опрос, практическая работа
10.1	Self, классы, методы объектов	16	6	10	

10.2	Метод __init__, переменные класса и объекта, метаклассы	14	4	10	
<b>11</b>	<b>Ввод - вывод</b>	<b>14</b>	<b>4</b>	<b>10</b>	
<b>12</b>	<b>Исключения</b>	<b>12</b>	<b>6</b>	<b>6</b>	опрос, практическая работа
<b>13</b>	<b>Стандартная библиотека</b>	<b>18</b>	<b>6</b>	<b>12</b>	
13.1	Константа	9	3	6	
13.2	Вспомогательные функции	9	3	6	
<b>14</b>	<b>Резерв</b>	<b>10</b>	<b>4</b>	<b>6</b>	
	<b>Итого</b>	<b>216</b>	<b>76</b>	<b>140</b>	

### Содержание учебно- тематического плана

#### 1. Вводное занятие.

**Теория:** Занятие проводится в форме беседы, экскурсии по учреждению. Демонстрация оборудования класса.

**Практика:** Запись правил.

#### 2. Блок – схемы.

**Теория:** Предпочтительнее до записи на алгоритмическом языке представить алгоритм в виде блок-схемы. Для построения алгоритма в виде блок-схемы необходимо знать назначения каждого из блоков.

**Практика:** Построение блок – схем.

##### 2.1 Функции блоков

**Теория:** Сколько всего блоков, какой за что отвечает, каким образом используется в проектировании, документировании и управления процессами или процедурами во многих областях

**Практика:** Запись.

##### 2.2 Линейный алгоритм.

**Теория:** Линейный алгоритм - это тип алгоритма, в котором последовательность действий не меняется в его процессе выполнения.

Линейные программы обычно предназначены для решения простейших задач, в которых не предусмотрен выбор из нескольких возможных направлений хода программы или циклическое повторение операций.

**Практика:** Решение математических задач с использованием линейного алгоритма «блок-схемы»

##### 2.3 Разветвляющийся алгоритм.

**Теория:** Разветвляющийся алгоритм – это алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.

Самый простой пример реализации разветвляющегося алгоритма – если на улице идет дождь, то необходимо взять зонт, иначе не брать зонт с собой. Приведенный выше пример псевдокода по нахождению частного двух чисел также относится к разветвляющемуся алгоритму.

**Практика:** Решение математических задач с использованием разветвляющейся алгоритма «блок-схемы»

## 2.4 Циклический алгоритм

**Теория:** Циклический алгоритм — алгоритм, предусматривающий многократное повторение одного и того же действия (одних и тех же операций) над новыми исходными данными. К циклическим алгоритмам сводится большинство методов вычислений, перебора вариантов. Цикл программы — последовательность команд (серия, тело цикла), которая может выполняться многократно (для новых исходных данных) до удовлетворения некоторому условию.

**Практика:** Решение математических задач с использованием циклического алгоритма «блок-схемы»

## 3. Основы программирования.

**Теория:** Во – первых, стоит отметить интересную особенность Python. Он не содержит операторных скобок (begin..end в pascal или {...} в Си), вместо этого блоки выделяются отступами: пробелами или табуляцией, а вход в блок из операторов осуществляется двоеточием. Однострочные комментарии начинаются со знака фунта «#», многострочные — начинаются и заканчиваются тремя двойными кавычками «"""».

Чтобы присвоить значение переменной используется знак «=», а для сравнения — «==». Для увеличения значения переменной, или добавления к строке используется оператор «+=», а для уменьшения — «-=». Все эти операции могут взаимодействовать с большинством типов, в том числе со строками.

**Практика:** Выполнение практических заданий.

### 3.1 Переменные и их виды.

**Теория:** Переменная — это простейшая именованная структура данных, в которой может быть сохранён промежуточный или конечный результат работы программы. 4 вида: целые числа, вещественные числа, логические значения, строковый тип.

**Практика:** Запись конспекта.

## 4. Операторы и выражения.



#### 4.1 Арифметические операторы.

**Теория:** Арифметические (математические) операторы выполняют математические операции над операндами и возвращают результат. В Python они такие же, как и в большинстве других языков программирования:

**Практика:** Выполнение практических заданий.

#### 4.2 Операторы сравнения.

**Теория:** Операторы сравнения в Python используются для сравнения двух объектов. Возвращаемый результат – логическое значение – True или False.

**Практика:** Выполнение практических заданий.

#### 4.3 Операторы присваивания.

**Теория:** Операторы присваивания в Python предназначены для присваивания значений переменным. Они также могут манипулировать значениями перед присваиванием. Простой оператор присваивания (=). Данный оператор присваивает значение.

**Практика:** Выполнение практических заданий.

#### 4.4 Логические операторы.

**Теория:** В Python есть три логических оператора, которые позволяют создавать сложные условия:

and — логическое умножение;

or — логическое сложение;

not — логическое отрицание.

**Практика:** Выполнение практических заданий.

#### 4.5 Операторы принадлежности.

**Теория:** Операторы принадлежности — это операторы, используемые для проверки принадлежности значения. Он проверяет принадлежность к последовательности, такой как строки, списки или кортежи. оператор in: оператор «in» используется для проверки того, существует ли значение в последовательности или нет.

**Практика:** Выполнение практических заданий.

#### 4.6 Операторы тождественности.

**Теория:** В Python существует два оператора тождественности: is и is not. Эти операторы сравнивают расположение двух объектов в памяти. Операторы тождественности используются не часто, но очень интересны с точки зрения того, как Python размещает объекты в памяти. Оператор "is". Возвращает. True, если оба операнда указывают на один и тот же объект, иначе возвращает False. Пример с числовыми переменными: x = 5 y = 5 a = x is y print (a) Получим True.

**Практика:** Выполнение практических заданий.

## 5. Поток команд.

**Теория:** в программах, которые мы до сих пор рассматривали, последовательность команд всегда выполнялась Python по порядку строго сверху вниз. А что, если нам необходимо изменить поток выполняющихся команд? Например, если требуется, чтобы программа принимала некоторое решение и выполняла различные действия в зависимости от ситуации; скажем, печатала «Доброе утро» или «Добрый вечер» в зависимости от времени суток. Как вы уже, наверное, догадались, этого можно достичь при помощи операторов управления потоком. В Python есть три оператора управления потоком: `if`, `for` и `while`.

**Практика:** Выполнение практических заданий.

### 5.1 Цикл `for`.

**Теория:** Python цикл `For`. Цикл `for` используется для перебора последовательности (то есть списка, кортежа, словаря, набора или строки). Это меньше похоже на ключевое слово `for` в других языках программирования и работает больше как метод итератора, который можно найти в других объектно-ориентированных языках программирования. С помощью цикла `for` мы можем выполнить набор операторов, один раз для каждого элемента в списке, кортеже, наборе и т. д.

**Практика:** Выполнение практических заданий.

### 5.2 Цикл `while`.

**Теория:** Цикл `while` используется в Python для неоднократного исполнения определенной инструкции до тех пор, пока заданное условие остается истинным. Этот цикл позволяет программе перебирать блок кода.

**Практика:** Выполнение практических заданий.

## 6. Функции.

### 6.1 Аргументы функции.

**Теория:** для объявления функции служит ключевое слово «`def`». Для вызова функции следует ввести ее имя и добавить скобки. Например: `my_function ("abracadabra")`. Аргументы функции в Python. Вызывая функцию, мы можем передавать ей следующие типы аргументов: Обязательные аргументы (Required arguments) Аргументы-ключевые слова (Keyword argument) Аргументы по умолчанию (Default argument) Аргументы произвольной длины (Variable-length arguments).

**Практика:** Выполнение практических заданий.

### 6.2 Анонимные функции, инструкция `lambda`

**Теория:** В Python анонимная функция – это функция, которая определяется без имени. В то время как обычные функции определяются с

помощью ключевого слова `def`, анонимные определяются с помощью ключевого слова `lambda`.

**Практика:** Выполнение практических заданий.

## 7. Модули.

**Теория:** Как можно использовать код повторно, помещая его в функции, мы уже видели. А что, если нам понадобится повторно использовать различные функции в других наших программах? Как вы уже, наверное, догадались, ответ – модули.

**Практика:** Выполнение практических заданий.

### 7.1 Модуль `CSV` – чтение и запись файлов.

**Теория:** Файл `CSV` – это особый вид файла, который позволяет структурировать большие объемы данных.

По сути, он является обычным текстовым файлом, однако каждый новый элемент отделен от предыдущего запятой или другим разделителем. Обычно каждая запись начинается с новой строки. Данные `CSV` можно легко экспортировать в электронные таблицы или базы данных. Программист может расширить `CSV` файл, добавляя новые строки.

**Практика:** Выполнение практических заданий.

### 7.2 Модуль `unittest`: тестируем свои программы.

**Теория:** Модуль `unittest` предоставляет богатый набор инструментов для написания и запуска тестов. Однако достаточно лишь некоторых из них, чтобы удовлетворить потребности большинства пользователей.

**Практика:** Выполнение практических заданий, тестируем свои программы.

## 8. Структуры данных

### 8.1 Список, кортеж, словарь.

**Теория:** Python содержит такие структуры данных как списки (`lists`), кортежи (`tuples`) и словари (`dictionaries`). Списки — похожи на одномерные массивы (но вы можете использовать Список, включающий списки — многомерный массив), кортежи — неизменяемые списки, словари — тоже списки, но индексы могут быть любого типа, а не только числовыми. "Массивы" в Python могут содержать данные любого типа, то есть в одном массиве могут находиться числовые, строковые и другие типы данных. Массивы начинаются с индекса 0, а последний элемент можно получить по индексу -1. Вы можете присваивать переменным функции и использовать их соответственно.

**Практика:** Выполнение практических заданий.

### 8.2 Последовательности, множества, ссылки.

**Теория:** можно использовать часть массива, задавая первый и последний индекс через двоеточие «:». В таком случае вы получите часть массива, от первого индекса до второго не включительно. Если не указан первый элемент, то отсчет начинается с начала массива, а если не указан последний — то массив считывается до последнего элемента. Отрицательные значения определяют положение элемента с конца.

**Практика:** Выполнение практических заданий.

## 9. Решение задач.

**Теория:** Обсудим план решения задач.

**Практика:** Мы рассмотрели различные части языка Python, и теперь посмотрим, как все эти части работают вместе, проектируя и составляя программу, которая делает что-то полезное. Цель состоит в том, чтобы научиться писать сценарии на языке Python самостоятельно.

## 10. Объектно-ориентированное программирование

### 10.1. Self, классы, методы объектов.

**Теория:** Язык Python ограничен в множественном наследовании в классах. Внутренние переменные и внутренние методы классов начинаются с двух знаков нижнего подчеркивания «\_\_» (например, «\_\_myprivatevar»). Мы можем также присвоить значение переменной класса извне.

**Практика:** Выполнение практических заданий.

### 10.2. Метод \_\_init\_\_, переменные класса и объекта, метаклассы.

**Теория:** Метод \_\_init\_\_ запускается, как только объект класса реализуется. Этот метод полезен для осуществления разного рода инициализации, необходимой для данного объекта. Обратите внимание на двойные подчёркивания в начале и в конце имени.

**Практика:** Выполнение практических заданий.

## 11. Ввод – вывод.

**Теория:** Ещё одним распространённым типом ввода/вывода является работа с файлами. Возможность создавать, читать и записывать в файлы является ключевой для многих программ, поэтому в настоящей главе и мы рассмотрим этот аспект.

**Практика:** Выполнение практических заданий.

## 12. Исключения.

**Теория:** Это касается и программ, содержащих недействительные команды.

В этом случае Python поднимает руки и сообщает, что обнаружил ошибку.

**Практика:** Выполнение практических заданий.

## 13. Стандартная библиотека.

### **13.1 Константа.**

**Теория:** рассмотрим некоторые наиболее часто используемые модули этой библиотеки. Детальное описание всех модулей стандартной библиотеки Python можно найти в разделе «Library Reference» документации, входящей в комплект поставки Python.

**Практика:** Выполнение практических заданий.

### **13.2 Вспомогательные функции.**

**Теория:** Вспомогательные функции, которые мы определяем внутри других функций. В Python такая функция имеет прямой доступ к переменным и именам, определенным во включающей её функции.

**Практика:** Выполнение практических заданий.

## **Планируемые результаты**

В результате освоения программы, обучающиеся будут обладать информационными компетенциями, и практическими навыками программирования на языке Python, а робототехника для решения типовых задач механики и конструирования.

***По окончании 1 года обучения обучающийся будет знать:***

- программу LEGO MINDSTORMS NXT;
- первоначальные знания по устройству робототехнических устройств;
- основные приемам сборки и программирования робототехнических средств;
- общенаучные и технологические термины и проектирования;
- правила безопасной работы с инструментами необходимыми при конструировании робототехнических средств.

***Будет уметь:***

- планировать результаты своей деятельности;
- строить алгоритм работы робота и программировать его;
- развивать умения излагать мысли в четкой логической последовательности, отстаивать свою точку зрения, анализировать ситуацию и самостоятельно находить ответы на вопросы путем логических рассуждений;

- развивать психофизиологические качества: память, внимание, способность логически мыслить, анализировать, концентрировать внимание на главном.

**В результате освоения программы 2 года обучения, обучающиеся будут знать:**

- самостоятельно решать технические задачи в процессе конструирования роботов;
- основы работы с программным обеспечением PyCharm
- создавать реально действующие модели роботов по собственному замыслу;
- создавать программы на компьютере для различных роботов;
- корректировать программы при необходимости;
- основы языка программирования Python;

***уметь:***

- принимать или намечать учебную задачу, ее конечную цель;
- прогнозировать результаты работы;
- планировать ход выполнения задания.;
- высказываться устно в виде рецензии ответа товарища.

***В результате обучения по программе обучающиеся приобретут такие личностные качества как:***

- самостоятельность и творческий подход к решению задач с помощью средств современной вычислительной техники.

- ответственное отношение к обучению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию.

- нравственные качества по отношению к окружающим (доброжелательность, взаимопомощь, уважение к труду окружающих);

***В результате обучения по программе у обучающихся будут сформированы такие метапредметные компетенции как:***

- психофизиологические качества обучающихся: память, внимание,

способность логически мыслить, анализировать, концентрировать внимание на главном;

- самостоятельно обнаруживать и формулировать учебную проблему, определять цель учебной деятельности;

- выдвигать версии решения проблемы, осознавать конечный результат, выбирать из предложенных и искать самостоятельно средства достижения цели;

- составлять (индивидуально или в группе) план решения проблемы (выполнения проекта);

- работая по плану, сверять свои действия с целью и, при необходимости, исправлять ошибки самостоятельно;

- умение искать и использовать необходимую информацию.

## **Комплекс организационно-педагогических условий**

### **Календарный учебный график**

Этапы образовательного процесса	1 год обучения	2 год обучения
Продолжительность учебного года	36 недель	36 недель
Количество учебных часов	216	216
Продолжительность занятия	3 по 40 мин	3 по 40 мин
Количество занятий в неделю	2	2
Начало учебного года	15 сентября	15 сентября
Окончание учебного года	31 мая	31 мая
Сроки комплектования учебных групп	1 сентября - 15 сентября	-
Режим работы в каникулярное время (осенние, зимние, весенние)	По расписанию Воспитательные мероприятия	По расписанию Воспитательные мероприятия
Режим работы в период летних каникул	Творческие смены, выезды	Творческие смены, выезды

## **Условия реализации программы**

Эффективность реализации программы зависит от условий, которые необходимо создать для организации образовательной деятельности.

### **Материально-техническое оснащение и средства обучения**

1. Lego Mindstorms NXT – 8 наборов;
2. Набор ресурсный средний – 4 набора;
3. Программное обеспечение NXT 2.0;
4. Датчики освещённости NXT – 8 шт;
5. Зарядные устройства – 8 шт;
6. Компьютер 8 шт.
7. Программное обеспечение Python
8. Программное обеспечение PyCharm
9. Проектор (интерактивная доска) доска

**Информационное обеспечение** Учебная и справочная литература. Выход в интернет, комплект видео материалов, методических разработок к основным разделам программы.

**Кадровое обеспечение.** Реализацию программы осуществляет педагог дополнительного образования, имеющий среднее образование (в том числе по направлению данной программы) и отвечающий квалификационным требованиям, указанным в Профессиональном стандарте «Педагог дополнительного образования детей и взрослых (Приказ Минтруда России от 05.05.2018 г. № 298н).

## **Формы контроля**

### **Способы отслеживания результатов реализации программы**

Наиболее подходящая форма оценки уровня освоения программы обучающимися – организованный просмотр выполненных работ в конце каждой темы, участие в соревнованиях также зачет, для получения которого нужно предоставить решения задач. В течение всего периода обучения проводятся наблюдения, опросы, беседы, анкетирование, как обучающихся,



так и их родителей. В течение каждого занятия обучающиеся задают вопросы друг другу по изученной теоретической и практической части программы в устной форме. Постоянно педагогом проводится индивидуальная работа с обучающимися по выполнению практической работы.

### **Оценочные материалы**

#### **Механизм отслеживания результатов:**

- выполнение тестов;
- диагностика уровня знаний за первое полугодие
- диагностика уровня освоения программы.
- выполнение тестовых задач;
- соревнования;
- учебно-исследовательские конференции;
- проекты;
- отзывы родителей учеников.

Все выявленные в процессе контроля недочеты, корректируются и устраняются. Упор делается на экспериментальную работу. На практические занятия выносятся перечень тех устройств разного направления, на базе которых можно рассмотреть изучаемые темы. Каждый ученик подбирает себе устройство того направления, которое его интересует. Но практическая часть занятий невозможна без понимания протекающих в них физических явлений и процессов, без умения производить простейшие расчеты, что требует изучения теоретических вопросов. Обучающиеся, сопоставляя свои знания с требованиями программы, дополняют, уточняют и расширяют их, создавая целостный компонент знаний, необходимый для сознательной и творческой работы в области робототехники и программирования.

Контролем усвоения материала и результатом обучения второго года обучения является:

- проект;
- изготовление работа в законченном виде, для определенных задач;
- участие в соревнованиях мобильных роботов работ;

- участие в итоговой выставке работ;
- внутри объединения во время школьных каникул проводятся соревнования по различным дисциплинам. Возможно ускоренное прохождение этого курса обучения.

### **Методические материалы**

При реализации образовательной программы предусматривается вариативность обучения, возможно опережающее развитие обучающихся, и поэтому темы занятий могут быть несколько изменены в соответствии со знаниями обучающихся, их способностями и умениями. Возможна корректировка тем. Содержание занятий дифференцировано с учетом возрастных и индивидуальных особенностей детей и подростков. В ней отражены условия для индивидуального творчества, а также для раннего личностного и профессионального самоопределения детей, их самореализации и саморазвития. Приведенный в программе перечень практических занятий является примерным и может быть изменен педагогом в зависимости от желаний, интересов обучающихся.

При реализации программы используются такие методы обучения: иллюстративно-объяснительный, частично-поисковый, проектный, метод практического применения знаний, метод проблемного обучения, форма группового творческого задания, дискуссия, беседа, дидактическая игра.

Формы работы: групповая и индивидуальная, а также различные типы компьютерных обучающих систем. Используются различные формы проведения занятий: беседа, лекция, демонстрация, занятие-соревнование, занятие-конкурс, викторина, КВН, турниры. Развивать творческие способности обучающихся помогает выполнение самостоятельных творческих работ, когда обучающийся сам придает работе индивидуальность и неповторимость.

На занятиях используются дидактические материалы (раздаточные материалы, инструкции, технологические карты, перечень стандартных

задач).

1. Стандартные задачи.
2. Конкурсные задачи.
3. Проекты по заданию.
4. Личные проекты.

## Список литературы

1. Бишоп, О. Настольная книга разработчика роботов [Текст] / О. Бишоп. – К.: «МК-Пресс, СПб. «КОРОНА-ВЕК», 2010. -400с., ил.
2. Брага, Н. Создание робота в домашних условиях / пер. с англ. Е. А. Добролежина [Текст] / Н. Брагина. – М.: НТ Пресс, 2007. – 368 с.: ил. –(Робот своими руками).
3. Программируемые роботы/ Дж. Вильямс; пер. с англ. А. Ю. Карцева [Текст].- М. : НТ Пресс, 2006. – 240 с. : ил. (Робот своими руками).
4. Сайт Lego Mindstorms NXT: робототехника для школ и ВУЗов Нижнего Новгорода (<http://nnxt.blogspot.ru> )
5. Филиппов, С.А. Робототехника для детей и родителей [Текст] / С. А. Филиппов. – СПб.: Наука, 2013. 319 с.
6. Юревич, Е. И. Основы робототехники [Текст] / Е. И. Юревич. – 2-е изд., перераб. и доп.– СПб.: БХВ – Петербург, 2005. – 416 с., ил.
7. Овсяницкая, Л.Ю. Курс программирования робота Lego Mindstorms EV3 в среде EV3 – Челябинск: ИП Мякотин И.В., 2014. – 204 с.
8. Овсяницкая Л.Ю. Алгоритмы и программы движения робота Lego Mindstorms EV3 по линии/ Л.Ю. Овсяницкая, Д.Н.Овсяницкая, А.Д. Овсяницкая-М.:Издательство «Перо», 2015.-168с.
9. Майк МакГрат «Программирование на Python для начинающих» Эксмо, 2015.
10. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. – Санкт-Петербург: 2016.
11. Сэнд У., Сэнд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.
12. Щерба А.В. Изучение языка программирования Python на основе задач УМК авторов И.А. Калинин и Н.Н. Самылкина. //Учебное пособие. – М.: МПГУ, 2015.

## Список рекомендуемой литературы для обучающихся

1. Брага, Н. Создание робота в домашних условиях / пер. с англ. Е. А. Добролежина [Текст] / Н. Брагина. – М.: НТ Пресс, 2007. – 368 с.: ил. – (Робот своими руками).
2. Сайт Lego Mindstorms NXT: робототехника для школ и ВУЗов Нижнего Новгорода (<http://nnxt.blogspot.ru> )
3. Филиппов, С.А. Робототехника для детей и родителей [Текст] / С. А. Филиппов. – СПб.: Наука, 2013. 319 с.
4. Устройства управления роботами (часть 1,2) [Текст]. - М. Предко пер. с англ. М. : ДМК, 2004.
5. Овсяницкая, Л.Ю. Курс программирования робота Lego Mindstorms EV3 в среде EV3 – Челябинск: ИП Мякотин И.В., 2014. – 204 с.
6. Овсяницкая Л.Ю. Алгоритмы и программы движения робота Lego Mindstorms EV3 по линии/ Л.Ю. Овсяницкая, Д.Н.Овсяницкая, А.Д. Овсяницкая-М.:Издательство «Перо», 2015.-168с.
7. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
8. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с.
9. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
10. Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 336 с.: ил. – (Серия «Бестселлеры O'Reilly»).