

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

Принята на заседании
педагогического совета
от 03 июня 2024 г.
Протокол №2



УТВЕРЖДАЮ:
Директор ГАУДО КЦ
Д.Ю. Ведягин
03 июня 2024 г.

Дополнительная общеобразовательная общеразвивающая программа
социально-гуманитарной направленности
«Робототехника и программирование»

базовый уровень
возраст обучающихся 12 – 17 лет
срок реализации 2 года



МЕЙКЕР

Разработчик:
Макаренко Данила Александрович,
педагог дополнительного образования

г. Кемерово, 2024 г.

Содержание

Комплекс основных характеристик программы.....	3
Пояснительная записка	3
Учебно-тематический план 1 год обучения.....	7
Содержание учебно-тематического плана 1 год обучения.....	11
Учебно-тематический план 2 год обучения.....	17
Содержание учебно-тематического плана 2 год обучения.....	20
Планируемые результаты	30
Комплекс организационно-педагогических условий.....	32
Календарный учебный график	32
Условия реализации программы	33
Методические материалы	35
Список использованной литературы и источников	37

1. 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 Российской Федерации от 29.12.2012 № 273 «Об образовании в Российской Федерации» (с изменениями и дополнениями);

– Изменения в Федеральный закон «Об образовании в Российской Федерации» 273-ФЗ в части определения содержания воспитания в образовательном процессе с 1.09.2020;

– Указа Президента Российской Федерации «О национальных целях развития Российской Федерации на период до 2030 года», определяющего одной из национальных целей развития Российской Федерации предоставление возможности для самореализации и развития талантов

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

– Стратегия развития воспитания в Российской Федерации на период до 2025 года, утвержденная Распоряжением Правительства Российской Федерации от 29 мая 2015 г. № 996-р.;

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

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

программам»;

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

– Приказ «Об утверждении Правил персонифицированного финансирования дополнительного образования детей в Кемеровской области – Кузбассе» Министерство образования Кузбасса №102 от 13.01.2023г;

– Приказ Минтруда России от 22.09.2021 № 652н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых»;

– 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".

Методические рекомендации:

– Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разно уровневые программы) (разработанные Минобрнауки России совместно с ГАОУ ВО «Московский государственный педагогический университет», ФГАУ «Федеральный институт развития образования», АНО ДПО «Открытое образование», 2015г.) (Письмо Министерства образования и науки РФ от 18.11.2015 № 09-3242);

– Методические рекомендации по реализации адаптированных дополнительных общеобразовательных программ, способствующих социально-психологической реабилитации, профессиональному самоопределению детей с ограниченными возможностями здоровья, включая детей-инвалидов, с учетом их особых образовательных потребностей. (Письмо Министерства образования и науки РФ № ВК-641/09 от 26.03.2016);

– Методические рекомендации по формированию механизмов обновления содержания, методов и технологий обучения в системе дополнительного образования детей, направленных на повышение качества

дополнительного образования детей, в том числе включение компонентов, обеспечивающих формирование функциональной грамотности и компетентностей, связанных с эмоциональным, физическим, интеллектуальным, духовным развитием человека, значимых для вхождения Российской Федерации в число десяти ведущих стран мира по качеству общего образования, для реализации приоритетных направлений научно-технологического и культурного развития страны Минпросвещения России письмо от 29.09.2023 №АБ 3935/06;

– Примерная программа воспитания. Утверждена на заседании Федерального учебно-методического объединения по общему образованию 2.06.2020 г.

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

Актуальность программы.

Робототехника стремительно развивается и проникает во все сферы жизни: промышленность, медицину, образование, развлечения и даже домашнее хозяйство. Спрос на специалистов по робототехнике растет, и образовательные программы, подобные этой, помогут удовлетворить этот спрос. Python стал одним из самых популярных языков программирования для робототехники благодаря своей простоте, гибкости и большому количеству библиотек, специально разработанных для работы с роботами. Обучая основам Python, даст пользователям ключевой навык для работы в этой области. Доступность и простота использования: Python – язык с низким порогом вхождения, что делает его идеальным инструментом для обучения. Программа, будучи ориентированной на обучение, может сделать робототехнику доступной для широкой аудитории, включая детей и подростков. Развитие навыков будущего: программирование и знание робототехники – ключевые навыки 21 века. Обучаясь по данной программе, ученики разовьют критическое мышление, логику, креативность и навыки решения проблем, которые пригодятся им в любой сфере деятельности.

Сочетание Python и робототехники открывает огромные возможности для инноваций.

Отличительной особенностью программы является ее содержание 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 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 12 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 программирования роботов на языке программирования Python 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 разделов\ тем	Количество часов:			Формы контроля
		Теория	Практика	Всего	
1.	Введение в образовательную программу, техника безопасности	3	-	3	---
1.1.	Вводное занятие. Техника безопасности	3	-	3	---
2.	Конструктор Lego Mindstorms Education EV3.	9	36	45	Решение задач программирования.
2.1.	Правила работы с конструктором LEGO Mindstorms Education EV3.	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 EV3.	1	5	6	Опрос
2.5.	Знакомство с датчиками.	1	11	12	Опрос
2.6.	Программирование во внутренней среде работа	3	9	12	Решение задач программирования
3.	Введение в механику	18	18	36	Упражнения, решение механических задач, проекты
3.1.	История механики, основные понятия и термины.	0	3	3	Упражнения, решение механических задач
3.2.	Робот для быстрого выполнения задач	0	3	3	Упражнения, решение механических задач

3.3.	Робот с повышенной проходимостью	0	3	3	Упражнения, решение механических задач
3.4.	Автономный робот для научных исследований	0	3	3	Упражнения, решение механических задач
3.5.	Робот с искусственным интеллектом для навигации	0	3	3	Упражнения, решение механических задач
3.6.	Персональный робот, предназначенный для индивидуального использования	3	18	21	Упражнения, проект
4.	Введение в Lego Mindstorms EV3 Home Education	15	21	36	Решение задач на программирование. Соревнования.
4.1.	Окно инструментов, Команды Home Education	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
5.	Блок движения	9	36	45	Выполнение упражнений, решение задач. Соревнования.

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.
6.	Блок ожидания	3	12	15	Выполнение упражнений, решение задач. Соревнования.
7.	Блок цикл	1	2	3	Выполнение упражнения, решение задачи.
8.	Блок переключатель	1	2	3	Выполнение упражнения, решение задачи
9.	Блок звука	1	2	3	Выполнение упражнения, решение задачи
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	-	---

	Итого:	66	150	216	---
--	---------------	-----------	------------	------------	------------

Содержание учебно-тематического плана 1-го года обучения

Раздел 1. Введение в образовательную программу, техника безопасности.

Тема 1.1. Вводное занятие. Техника безопасности.

Теория. Введение в образовательную программу «Робототехника и программирование». Техника безопасности на рабочем месте. Что такое робот. Робот и современность. Виды языков программирования. Виды роботов. Перспективы развития робототехники и программирования.

Практика. Создание мини-проекта: Каждая группа из 2-3 человек разрабатывает краткое описание робота, включая его функции и использование (например, робот-пылесос, промышленный робот).

Раздел 2. Конструктор Lego Mindstorms Education EV3.

Тема 2.1. Правила работы с конструктором Lego Mindstorms Education EV3.

Теория. Знакомство с набором Lego Mindstorms Education EV3, техника безопасности и правила при использовании конструктора Lego Mindstorms Education EV3.

Практика. Сборка робота: Учащиеся собирают робота по инструкции, работая в команде, распределяя роли (один собирает детали, другой контролирует процесс).

Тема 2.2. Основные детали.

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

Практика. Каждая группа получает задание собрать простую модель (например, автомобиль, манипулятор или мост), используя все изученные детали. Группа должна взять за основу функциональность конструкции:

- Использовать балки для создания рамы.
- Включить оси для подвижных частей.
- Использовать захваты для манипуляций.
- Применять втулки для осей и фиксаторы для стабилизации модели.

Тема 2.3. Спецификация кнопки управления.

Теория. Кнопки управления EV3 Оранжевая: включить/Ввести. Светло-серые стрелки: Навигация, влево и Вправо. Тёмно-серая: очистить/Вернуться.

Практика. Напишите краткий отчет о том, как кнопки управления помогают в процессе программирования и управления роботом EV3.

Тема 2.4. Назначение портов EV3.

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. Потренируемся в подключении: подключим к роботу: Мотор к порту "А". Датчик цвета к порту "В" * Проверим, что устройства подключены правильно (лампочка на датчике должна загореться). Потренироваться в подключении по Bluetooth: включите Bluetooth на вашем компьютере. Включите Bluetooth на роботе EV3. Найдите робот EV3 в списке доступных устройств на вашем компьютере. Сопрягите робот с вашим компьютером. Проверим, что подключение установлено (индикатор Bluetooth на роботе должен мигать).

Тема 2.5. Знакомство с датчиками.

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

Практика. Использование каждого датчика. Попробуем использовать датчики в комбинации друг с другом для создания более сложных программ. Попробуем реализовать свои собственные идеи, используя датчики EV3.

Тема 2.6. Программирование во внутренней среде.

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

Практика. Блоки движения: используем блок "Движение" для управления двигателями робота. Заставим робота двигаться вперед, назад, поворачивать на месте. Изменяем скорость и время движения робота. Используем блок "Свет" для измерения уровня освещенности. Заставьте

робота двигаться к источнику света. Заставьте робота останавливаться, когда уровень освещенности достигает определенного значения. Блоки звука: используем блок "Звук" для воспроизведения звуков. Заставьте робота издавать звуковой сигнал, когда он сталкивается с препятствием. Заставьте робота воспроизводить мелодию.

Блоки времени: Используйте блок "Время" для управления временем выполнения программы. * Заставьте робота ждать определенное количество времени. Заставьте робота выполнять действия с определенной периодичностью.

Раздел 3. Введение в механику.

Тема 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, опрос, кроссворд.

Тема 3.2. Робот для быстрого выполнения задач.

Теория. Изучение деталей для конструирования.

Практика. Конструирование робота для быстрого выполнения задач.

Тема 3.3. Робот с повышенной проходимостью.

Теория. Изучение деталей для конструирования.

Практика. Конструирование робота внедорожника.

Тема 3.4. Автономный робот для научных исследований.

Теория. Изучение деталей для конструирования.

Практика. Конструирование робота исследователя.

Тема 3.5. Робот с искусственным интеллектом.

Теория. Изучение деталей для конструирования.

Практика. Конструирование робота автобота.

Тема 3.6. Персональный робот, предназначенный для индивидуального использования.

Теория. Подбор деталей для своего робота, составление плана,

придумывание названия.

Практика. Конструирование своего робота.

Раздел 4. Введение в Lego Mindstorms EV3 Home Education.

Тема 4.1. Окно инструментов, команды Home Education.

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

Практика. выполняем задание:

1. Запуск программы: Откройте программу Home Education.
2. Изучение интерфейса: Найдите окно инструментов, где расположены основные элементы управления (например, кнопки "Создать", "Открыть", "Сохранить"). Изучите разделы меню (например, "Файл", "Правка", "Вид"). Попробуйте использовать разные инструменты и команды, чтобы понять их назначение.

3. Создание профиля: Нажмите на кнопку "Создать" или выберите "Создать профиль" в меню "Файл". Введите имя своего профиля, укажите свой возраст и другие необходимые данные. Сохраните профиль.

Тема 4.2. Работа с пиктограммами, соединение команд.

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

Практика. Создайте свою уникальную программу, применяя другие смешанные пиктограммы. Поделитесь своим проектом с группой и объясните использование каждой пиктограммы.

Тема 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.

1. Создание линейной программы: Откройте программу Home Education. Создайте новый проект. Добавьте в проект блок "Движение" и настройте его

для движения робота вперед на 1 секунду. Добавьте блок "Звук" и настройте его для воспроизведения звукового сигнала после окончания движения.

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

Теория. Эта тема посвящена основам программирования: как создавать программы с параметрами, использовать циклы, проверять условия, управлять потоком выполнения с помощью условных переходов и работать с датчиками и их параметрами.

Практика. Выполнение практических задач с использованием цикла, цикла внутри цикла, условного перехода и их параметрами.

Раздел 5. Блок движения.

Тема 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. Выполнение практических задач.

Тема 5.2. Программирование автономного робота на объезд препятствий.

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

Практика. Создание программы на объезд ученика роботом и запуск.

Тема 5.3. Выполнение блока движения при определённых условиях.

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

Практика. Программирование блока на движение при определённых условиях.

Раздел 6. Блок ожидания.

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

Практика. Выполнение задания: Создание программы, которая заставляет робота двигаться вперед на 1 секунду, затем ожидает 2 секунды, а затем двигается назад на 1 секунду. Включение в программу блока «Движение» для управления движением робота. Добавление блока «Ожидание» и настройка его на 2 секунды.

Раздел 7 Блок цикл.

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

Практика. 1. Создание простого цикла: Реализуйте цикл, который выведет на экран числа от 1 до 10.

2. Условие выхода: Напишите цикл, который будет запрашивать у пользователя ввод числа и завершится, когда пользователь введет "0".

3. Цикл с массивами: Создайте массив из 5 элементов и с помощью цикла найдите и выведите их сумму.

Раздел 8 Блок переключатель.

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

Практика. Выполнение практических задач:

1. Простой переключатель: Реализуйте переключатель, который будет определять день недели по номеру (1-7) и выводить его название.

2. Управление режимами: Напишите программу для робота, использующую переключатель для выбора режима работы (например, «Поиск», «Сбор», «Доставка»).

3. Обработка пользователей: Создайте программу, где переключатель будет обрабатывать команды от пользователя, например, «Старт», «Стоп» и «Пауза».

Раздел 9 Блок звука.

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

Практика. Выполнение практических задач:

1. Производство звука: Напишите программу, которая воспроизводит звук (например, мелодию) при нажатии кнопки.

2. Звуковые уведомления: Реализуйте систему, которая будет издавать звуковой сигнал при выполнении определённых действий робота (например, при завершении задачи).

3. Регулировка громкости: Создайте интерфейс, где пользователь может регулировать громкость звука с помощью слайдера.

Раздел 10. Блок дисплея.

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

Практика. Выполнение практических задач:

1. Отображение текста: Создайте программу, которая отображает приветственное сообщение на дисплее при включении робота.

2. Динамическое обновление: Разработайте проект, где дисплей показывает текущую температуру или уровень заряда батареи, обновляясь через заданные промежутки времени.

3. Графические элементы: Реализуйте отображение простых графиков или иконок на дисплее в зависимости от состояний робота (например, состояние движения/паузы).

Раздел 11. Блок записи/воспроизведения.

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

Практика. Выполнение практических задач:

1. Запись действий: Реализуйте функцию, которая записывает последовательность движений робота в течение заданного времени.

2. Воспроизведение записи: Создайте программу, которая воспроизводит записанные действия робота, позволяя ему автоматически повторять маршрут.

3. Редактирование записей: Разработайте возможность изменения записанных команд перед воспроизведением, например, добавление или

удаление шагов.

Раздел 12. Проект на заданную тему.

Теория. Постановка задачи проекта условия выполнения, ограничения.

Выяснение всех непонятных вопросов.

Практика. Выполнение проекта.

Раздел 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	
1.	Вводное занятие, техника безопасности	3	3	0	устный опрос
2.	Блок – схемы	12	4	8	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	
3.	Основы программирования	9	3	6	опрос, практическая работа
3.1.	Переменные и их виды	9	3	6	
4.	Операторы и выражения	6	2	4	опрос, практическая работа
4.1.	Арифметические операторы	1		1	
4.2.	Операторы сравнения	1		1	
4.3.	Операторы присваивания	1		1	опрос, практическая работа

4.4.	Логические операторы	1		1	опрос, практическая работа
4.5.	Операторы принадлежности	1		1	опрос, практическая работа
4.6.	Операторы тождественности	1		1	опрос, практическая работа
5.	Поток команд	15	6		опрос, практическая работа
5.2.	Цикл for	6	3	3	опрос, практическая работа
5.3.	Цикл while	9	3	6	опрос, практическая работа
6.	Структуры данных	30	10	20	опрос, практическая работа
6.1.	Список, кортеж, словарь	12	6	6	опрос, практическая работа
6.2.	Последовательность и, множества, ссылки	18	4	14	опрос, практическая работа
7.	Программирование сервоприводов	12	3	9	опрос, практическая работа
7.1.	Настройка сервоприводов	3	1	2	опрос, практическая работа
7.2.	Измеряемые параметры	3	1	2	опрос, практическая работа
7.3.	Движения и остановки	6	1	5	опрос, практическая работа
8.	Программирование Датчиков	30	10	20	опрос, практическая работа
8.1.	Датчик цвета	12	6	6	опрос, практическая работа
8.2.	Ультразвуковой датчик	18	4	14	опрос, практическая работа
8.3.	Датчик касания				опрос, практическая работа

9.	Решение задач	16	4	12	опрос, практическая работа
10.	Объектно – ориентированное программирование	30	10	20	опрос, практическая работа
10.1.	Self, классы, методы объектов	16	6	10	опрос, практическая работа
10.2.	Метод __init__, переменные класса и объекта, метаклассы	14	4	10	опрос, практическая работа
11.	Ввод - вывод	14	4	10	опрос, практическая работа
12.	Исключения	12	6		опрос, практическая работа
13.	Стандартная библиотека	18	6	12	
13.1.	Константа	9	3	6	опрос, практическая работа
13.2.	Вспомогательные функции	9	3	6	опрос, практическая работа
14.	Резерв	10	4	6	
	Итого	216	76	140	

Содержание учебно- тематического плана

Раздел 1. Вводное занятие.

Теория. Занятие проводится в форме беседы, экскурсии по учреждению.

Демонстрация оборудования класса.

Практика. Запись правил.

Раздел 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.

Тема 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.

Тема 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»

Тема 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 разветвляющегося алгоритма «блок-схемы»

Тема 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»

Раздел 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.

Тема 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.

Раздел 4. Операторы и выражения.

Тема 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. Задание на предоставление применения арифметических операторов в повседневной жизни.

Тема 4.2. Операторы сравнения.

Теория. Операторы сравнения в Python используются для сравнения

двух объектов. Возвращаемый результат – логическое значение – True или False.

Практика. напишем код, который: запрашивает у пользователя два числа, сравнивает эти числа с помощью операторов сравнения, выводит результат сравнения на экран.

Написать код, который: запрашивает у пользователя число. проверяет, является ли это число четным или нечетным, выводит результат на экран. В случае ошибок – найти их, проанализировать и сделать вывод.

Тема 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. напишем код, который: создаёт переменную `z` и присваивает ей значение `10`. * делит `z` на `2` (используя `/=`). * выводит новое значение `z` на экран. * Вычитает из `z` число `3` (используя `-=`). * выводит новое значение `z` на экран.

Тема 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.

`and` возвращает `True`, только если оба операнда `True`. `or` возвращает `True`, если хотя бы один операнд `True`. `not` меняет логическое значение операнда на противоположное.

Практика. Выполнение задание:

1. Запросите у пользователя его возраст.
2. Используйте логические операторы для проверки следующих условий: * `and`: возраст пользователя находится в диапазоне от 18 до 65 лет (включительно). * `or`: возраст пользователя меньше 18 лет или больше 65 лет.

* `not`: возраст пользователя не равен 21 году.

3. Выведите на экран соответствующее сообщение в зависимости от результата проверки.

Тема 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. написать код, который: запрашивает у пользователя возраст и рост, проверяет, является ли человек взрослым (старше 18 лет) и высоким (рост выше 180 см), выводит соответствующее сообщение на экран.

Написать код, который: запрашивает у пользователя день недели (1-7). Проверяет, является ли день выходным (суббота или воскресенье) и выводит соответствующее сообщение на экран.

Тема 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. создать список `'list1'` и присвоить ему несколько элементов.
* Создать второй список `'list2'`, присвоив ему `'list1'` (например, `'list2 = list1'`).
* Проверить, тождественны ли `'list1'` и `'list2'`, используя `'is'`. * Изменить один из элементов в `'list1'`. * Проверить, тождественны ли `'list1'` и `'list2'` снова. * Объяснить результаты, используя концепцию передачи ссылки на список.

Обсудить результаты: проверить правильность выполнения заданий. Обсудить, как результаты зависят от типа данных, как Python обрабатывает

ссылки на объекты в памяти. * Объяснить, когда целесообразно использовать оператор `is` и когда лучше использовать оператор `==` для сравнения.

Раздел 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. напишем код, который: запрашивает у пользователя число. Проверяет, является ли число положительным, отрицательным или нулем. Выводит соответствующее сообщение на экран.

Напишем код, который: запрашивает у пользователя два числа. Проверяет, является ли первое число больше второго. Выводит на экран сообщение о том, какое число больше.

Напишем код, который: запрашивает у пользователя день недели (1-7). Выводит на экран сообщение о том, какой сегодня день недели. Использует оператор `if` с несколькими условиями (`elif`) для определения дня недели.

Тема 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.

Вот несколько примеров того, как цикл `for` работает в реальной жизни:

Игра в прятки: Ты прячешься, а твой друг должен тебя найти. Он может ходить по одной комнате, потом по другой, и так пока не найдет тебя. Цикл `for` может помочь ему пройти по всем комнатам, чтобы точно тебя найти.

Рисование: Ты хочешь нарисовать 10 цветочков. Ты можешь нарисовать один, потом второй, и так 10 раз. Цикл `for` может нарисовать все 10 цветочков за тебя!

Чтение книги: Представь, что ты читаешь книгу по одной странице. Ты можешь прочитать первую, потом вторую, и так до конца книги. Цикл `for` может прочитать все страницы за тебя!

Практика. предоставить ещё несколько примеров использования цикла в повседневной жизни и написать код для своего же примера. Например:

1. Рисуем звездочки: Нарисуй 5 звездочек на листе бумаги.
2. Считаем до 10: Скажи вслух числа от 1 до 10.
3. Хлопаем в ладоши: Хлопни в ладоши 3 раза.

Тема 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. В реальной жизни он выглядит так: Ты хочешь убрать свою комнату. Ты собираешь игрушки, складываешь книги, убираешь мусор. Цикл `while` говорит: "Убирать, убирать, убирать..." пока ты не уберешь всю комнату. Т.е. выполнять действие, пока не выполнится определённое условие.

Практика. Игра "Угадай число": Ты загадываешь число от 1 до 10. Друг пытается угадать это число. Ты говоришь ему, больше или меньше загаданное число. Друг продолжает угадывать, пока не угадает правильное число. Надо использовать цикл `while` для проверки, угадал ли друг число.

Раздел 6. Структуры данных.

Тема 6.1. 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 может 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. Выполним задания:

1. Создание списков и работа с элементами: Создайте список с названиями дней недели. Добавьте в список новый день (например, "Выходной"). Удалите из списка один из дней. Выведите на экран все дни из списка с помощью цикла `for`. * Выведите на экран только первые три дня из списка. Выведите на экран последний день из списка.

2. Кортежи: Создайте кортеж с именами членов вашей семьи. Попробуйте изменить имя одного из членов семьи в кортеже. Что произошло? Выведите на экран все имена из кортежа с помощью цикла `for`.

3. Словари: Создайте словарь, где ключами будут имена друзей, а значениями - их номера телефонов. Добавьте в словарь нового друга с его номером телефона. Измените номер телефона одного из друзей. Выведите на экран все имена друзей из словаря с помощью цикла `for`. Выведите на экран номер телефона одного из друзей по его имени.

Тема 6.2. 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. создаём массив с числами от 1 до 10. Выводим на экран

элементы массива с 3-го по 6-й (включительно). Выводим на экран элементы массива с 1-го до последнего. Выводим на экран элементы массива с 3-го до предпоследнего. Выводим на экран элементы массива, начиная с 5-го. Выводим на экран элементы массива, заканчивая 7-м.

2. Срезы с шагом: создаём массив с числами от 1 до 10. Выведите на экран элементы массива с шагом 2 (т.е. 1-й, 3-й, 5-й и т.д.). Выведите на экран элементы массива в обратном порядке с шагом 2.

3. Срезы с отрицательными индексами: создаём массив с числами от 1 до 10. Выведите на экран элементы массива с 3-го по предпоследний (используя отрицательные индексы). Выведите на экран 3 последних элемента массива.

4. Модификация срезов: создаём массив с числами от 1 до 10. Замените элементы массива с 3-го по 5-й на нули. Добавьте в массив с 3-го по 5-й элементы из нового массива [11, 12, 13].

5. Практика со строками: создаём строку "Hello, world!". Выведите на экран подстроку с 7-го по 12-й символ. Выведите на экран первые 5 символов строки. Выведите на экран последние 5 символов строки. Выведите на экран каждый второй символ строки.

Раздел 7. Программирование сервоприводов.

Теория. Как можно использовать код, чтобы ввести в действие сервомоторы импортируя его из модуля `pybricks.ev3devices`. Изучим 3 аргумента, проверим работоспособность и выполним несколько практических заданий.

Практика. Подключение и проверка работоспособности. Подключим сервомотор к одному из портов EV3 Brick. Импортируем необходимые модули: создаём объект сервомотора, указав порт подключения и проверяем работоспособность сервомотора.

Тема 7.1. Настройка сервоприводов.

Теория. Теперь, когда мы умеем создавать объект сервопривода, можно попробовать воспользоваться его настройками. Начнем мы с методов,

затрагивающих текущее состояние мотора. Метод `angle()` и метод `speed()`:

Практика. Используем метод `angle()` для получения текущего угла сервомотора и используем метод `speed()` для получения текущей скорости сервомотора.

Тема 7.2. Изменяемые параметры.

Теория. и последним остаётся изучить параметры сервомоторов, которые поддаются изменениям и могут меняться в процессе выполнения работы роботом.

Практика. В этой практике мы изучим методы, которые позволяют изменять параметры сервопривода в процессе выполнения программы: * `set_run_settings(speed, acceleration, deceleration)`: устанавливает настройки движения сервопривода. * `speed`: максимальная скорость вращения в градусах в секунду. * `acceleration`: ускорение в градусах в секунду в квадрате. * `deceleration`: замедление в градусах в секунду в квадрате.

Тема 7.3. Движения и остановки.

Теория. Движение робота можно запрограммировать по-разному, светло, темно, видит красный, видит синий и т.д. Остановиться тоже по разным причинам можно, на данном занятии будет изучением способов движения и остановки

Практика. В этой практике мы создадим несколько сценариев движения робота, основываясь на данных от разных датчиков.

Раздел 8. Программирование датчиков.

Теория. Понятие датчиков, их типы и их применение в различных, обучимся их программировать и калибровать.

Практика. В этой практике мы изучим основные типы датчиков, используемых в Pybricks, и выполним ряд практических заданий, чтобы освоить их программирование и калибровку.

Тема 8.1. Датчик цвета.

Теория. Изучение всех возможностей датчика цвета

Практика. Выполнение практических заданий. И выполнить задание:

робот двигается, пока не увидит определенный цвет (например, красный). Когда видит нужный цвет, издает звук или загорается светодиодом. Можно добавить условие: если цвет не тот, то робот должен повернуться и продолжить движение.

Тема 8.2. Ультразвуковой датчик.

Теория. Изучение всех возможностей ультразвукового датчика

Практика. Выполнение практических заданий. Выполнение практических заданий. И выполнить задание: робот двигается, пока не увидит стену на определённом расстоянии (например, 25см). Когда видит, издает звук или загорается светодиодом. Можно добавить условие: если цвет не тот, то робот должен повернуться и продолжить движение.

Раздел 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.

Раздел 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. Создаём класс `Animal` с атрибутом `name` и методом `make_sound`. Создаём два класса `Dog` и `Cat`, наследующие от `Animal`. Добавьте в класс `Dog` внутреннюю переменную `_breed` и метод `_bark`, который выводит сообщение "Гав!". Добавьте в класс `Cat` внутреннюю переменную `_color` и метод `_meow`, который выводит сообщение "Мяу!". Создаём экземпляр класса `Dog` и `Cat`. Попробуйте получить доступ к внутренним переменным и методам классов (`_breed`, `_color`, `_bark`,

`_meow`)). Измените значение атрибута `name` у созданных экземпляров.

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. Создайте класс `Car` с атрибутами `brand`, `model`, `year` и `color`. Добавьте в класс `Car` метод `__init__`, который принимает эти атрибуты в качестве аргументов и инициализирует их для каждого нового объекта класса. Создайте несколько экземпляров класса `Car` с разными значениями атрибутов. Выведите значения атрибутов для каждого созданного объекта.

Раздел 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. Создайте текстовый файл `data.txt`. Напишите программу, которая записывает в файл `data.txt` текст "Привет, мир!". Напишите программу, которая читает содержимое файла `data.txt` и выводит его на экран. Добавьте в файл `data.txt` строку "Это вторая строка". Напишите программу, которая удаляет файл `data.txt`.

Раздел 12. Исключения.

Теория. Это касается и программ, содержащих недействительные команды.

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

Практика. Напишем программу, которая запрашивает у пользователя число и делит 10 на это число. Добавим блок `try-except`, который будет перехватывать ошибку деления на ноль (`ZeroDivisionError`). В блоке `except`

выведем сообщение об ошибке. Попробуйте ввести число 0 и убедитесь, что программа выведет сообщение об ошибке, а не завершится с ошибкой.

Раздел 13. Стандартная библиотека.

Тема 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. 1. ``math``: Используйте модуль ``math`` для вычисления квадратного корня, синуса и косинуса числа. Выведите результаты на экран.

2. ``random``: Используйте модуль ``random`` для генерации случайного числа в диапазоне от 1 до 10. Выведите сгенерированное число на экран.

3. ``datetime``: Используйте модуль ``datetime`` для получения текущей даты и времени. Выведите полученную дату и время на экран.

Тема 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 задания:

1. Создайте функцию ``calculate_total()``, которая принимает два числа в качестве аргументов.

2. Внутри ``calculate_total()`` создайте вложенную функцию ``square()``, которая принимает число и возвращает его квадрат.

3. В ``calculate_total()`` вызовите ``square()`` для каждого из входных чисел.

4. Сложите полученные квадраты и верните результат.

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 Ev3;
- 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	16 сентября	16 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 Ev3 – 8 наборов;
2. Micro SD на 1 гб – 8 штук
3. Адаптер Micro SD – 8 штук
4. Набор ресурсный средний – 4 набора;
5. Программное обеспечение Lego Mindstorms Home Edition ;
6. Зарядные устройства – 8 шт;
7. Компьютер 8 шт.
8. Программное обеспечение Python
9. Программное обеспечение PyCharm
10. 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 робота 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»).