

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ  
ГУМАНИТАРНО-ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО ЮУРГПУ)  
ТЕХНОПАРК УНИВЕРСАЛЬНЫХ ПЕДАГОГИЧЕСКИХ КОМПЕТЕНЦИЙ

СОГЛАСОВАНО:  
Директор технопарка УПК  
*А.В. Ворожейкина*  
«03» *09* 2025 г.

УТВЕРЖДАЮ:  
Ректор ФГБОУ ВО «ЮУРГПУ»  
*А.И. Кузнецов*  
«03» 2025 г.  
М.П.



Дополнительная общеобразовательная (общеразвивающая) программа  
технической направленности

«Информатика. Программирование и робототехника»

Челябинск, 2025

## РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

### 1. Пояснительная записка

Программа «Информатика. Программирование и робототехника» является дополнительной общеобразовательной общеразвивающей программой технической направленности и предназначена для дополнительного образования обучающихся 8–9 классов общеобразовательных организаций, проявляющих интерес к IT-сфере, инженерному делу и STEM-направлениям. Программа реализуется в рамках проектного взаимодействия Технопарка универсальных педагогических компетенций ФГБОУ ВО ЮУрГГПУ со школами города и области.

Программа разработана в соответствии с нормативно-правовыми документами:

- Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации» (статья 34);
- Приказ Министерства просвещения РФ от 09.11.2018 № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам» (зарегистрирован 29.11.2018 № 52831);
- Приказ Министерства просвещения РФ от 27 июля 2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- СанПиН 2.4.4. 3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей»;
- Письмо Минобрнауки России от 18.11.2015 № 09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ»);
- Приказ Минобрнауки РФ от 05.03.2004 № 1089 «Об утверждении Федерального компонента государственных стандартов начального общего, основного общего и среднего (полного) общего образования».

## **2. Актуальность программы и новизна**

Актуальность программы обусловлена стремительной цифровизацией экономики и высоким спросом на специалистов в области разработки программного обеспечения, автоматизации и робототехники. Программа позволяет школьникам 8–9 классов получить прикладные навыки алгоритмизации, работы с микроконтроллерами и датчиками, а также сформировать инженерное мышление в условиях проектной и лабораторной деятельности.

Новизна программы заключается в реализации интегрированного подхода, объединяющего программную разработку (Python, C++) и аппаратную робототехнику. Образовательный процесс выстроен на принципах *problem-based learning* и конструкторского проектирования, что обеспечивает естественную преемственность от базовых алгоритмов к созданию автономных интеллектуальных систем и подготовке к профильным олимпиадам и техническим соревнованиям.

## **3. Цели и задачи программы**

**Основная цель:** формирование устойчивых компетенций в области алгоритмизации, программирования и робототехники, развитие инженерно-технического мышления и подготовка обучающихся к осознанному выбору профиля обучения и будущей IT/инженерной профессии.

### **Задачи:**

#### *Обучающие:*

- освоить базовые и продвинутые конструкции языков программирования (Python, C++);
- изучить принципы работы микроконтроллеров, датчиков, исполнительных механизмов и схем подключения;
- научиться отлаживать код, собирать и программировать робототехнические платформы;
- освоить методы проектной деятельности и техническую документацию.

*Развивающие:*

- развитие алгоритмического, логического и системного мышления;
- формирование навыков моделирования физических и программных процессов;
- развитие пространственного воображения, технической интуиции и способности к нелинейному решению задач;
- развитие навыков работы с измерительным оборудованием и программными симуляторами.

*Воспитательные:*

- воспитание ответственности, аккуратности, культуры безопасной работы с электроникой и инструментами;
- формирование навыков командного взаимодействия, умения презентовать технические решения и аргументированно отстаивать инженерные решения;
- воспитание устойчивого интереса к научно-техническому творчеству и готовности к непрерывному саморазвитию в быстро меняющейся технологической среде.

#### **4. Планируемые результаты освоения программы**

**Личностные результаты** обеспечивают ценностно-смысловую ориентацию обучающихся: сформированность мотивации к освоению IT-профессий, понимание роли технологий в современном обществе, осознание этических и правовых норм в цифровой среде, умение соотносить технические решения с требованиями безопасности и экологичности.

**Регулятивные УУД** обеспечивают организацию учебной и проектной деятельности:

- **целеполагание:** постановка учебной/проектной задачи на основе анализа условий и ограничений;
- **планирование:** определение последовательности этапов разработки, подбор аппаратных и программных средств;

- прогнозирование: предвидение возможных ошибок в коде или схемотехнике, оценка временных затрат;
- контроль и коррекция: сравнение фактических результатов работы устройства/программы с заданным алгоритмом, внесение изменений в код или конструкцию;
- оценка: осознание уровня усвоения материала, рефлексия эффективности выбранных решений;
- волевая саморегуляция: способность преодолевать технические трудности, настойчивость в отладке сложных систем.

**Познавательные УУД** включают общенаучные, логические действия и действия по постановке и решению проблем:

- самостоятельное выделение и формулирование познавательной цели;
- поиск и структурирование технической информации, работа с документацией API, datasheet;
- знаково-символические действия: построение блок-схем, схем подключения, UML-диаграмм, алгоритмов;
- моделирование: создание программных и физических моделей роботов, проведение виртуальных экспериментов;
- логические действия: анализ условий задачи, синтез алгоритма, выделение причинно-следственных связей в работе датчиков и приводов;
- постановка и решение проблем: самостоятельное создание способов отладки, оптимизации кода и конструкции.

**Коммуникативные УУД** обеспечивают социальную компетентность и учет позиций других людей:

- планирование учебного сотрудничества с преподавателем и сверстниками, распределение ролей в проектной группе;
- постановка вопросов, инициативное сотрудничество в поиске технических решений;

- разрешение конфликтов: выявление технических противоречий, поиск альтернативных архитектурных решений, принятие консенсуса;
- управление поведением партнера: контроль, коррекция, оценка действий участников группы;
- умение точно и полно выражать мысли в устной и письменной форме (технические отчеты, презентации, защита проектов), владение профессиональной терминологией.

**Категория обучающихся:** обучающиеся 14–15 лет (8–9 классы).

**Требования к уровню подготовки:** без специальных требований.

**Количество человек в группе:** 10–15.

**Форма обучения:** очная.

**Формы организации учебной деятельности:** групповая, проектная, лабораторная, фронтальная.

**Режим занятий:** 1 раз в неделю блоком по 3 академических часа (или 3 раза в неделю по 1 академическому часу).

**Объем программы:** 192 академических часа (96 часов в год).

## 2. СОДЕРЖАНИЕ ПРОГРАММЫ

### 2.1. Учебный план

№ п/п	Наименование разделов (модулей) и тем	Всего ауд. часов (акад. час)	Теоретические занятия	Практические занятия
1.	Алгоритмы и основы Python	18	8	10
2.	Структурное программирование и работа с данными	18	6	12
3.	Аппаратная база робототехники (Arduino/Micro:bit)	16	6	10
4.	Датчики, моторы и автоматизация процессов	24	8	16
5.	Объектно-ориентированное программирование (ООП)	24	8	16
6.	Беспроводные технологии и Интернет вещей (IoT)	20	6	14
7.	Компьютерное зрение и элементы искусственного интеллекта	20	6	14
8.	Проектный модуль и подготовка к соревнованиям	28	6	22
9.	Итоговая аттестация и защита проектов	24	6	18
	<b>Итого</b>	<b>192</b>	<b>60</b>	<b>132</b>

### 2.2. Календарный учебный график

Продолжительность освоения программы	2 учебных года (64 недели; 32 недели – 8 класс; 32 недели – 9 класс)
Периодичность проведения занятий	Еженедельно (3 академических часа)
Входной контроль	Тестирование + творческое задание на алгоритмику
Промежуточная аттестация (конец 1 года)	Практический зачет + мини-проект
Итоговая аттестация	Защита комплексного робототехнического/программного проекта

### 2.3. КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

## 8 КЛАСС (96 часов, 32 недели)

№ п/п	Наименование разделов и тем	Кол-во часов	Основные виды деятельности обучающихся	Формы контроля
	Модуль 1. Введение в алгоритмизацию и Python	24		
1	Лекция. Введение в курс. Техника безопасности. Знакомство со средой разработки Python	3	– изучать правила ТБ в кабинете информатики; устанавливать и настраивать IDE (PyCharm/Thonny); запускать первые программы «Hello World»	Входной инструктаж
2	Практическая работа. Переменные, типы данных, ввод-вывод информации	3	– объявлять переменные различных типов; использовать функции <code>input()</code> , <code>print()</code> ; выполнять преобразование типов данных	Практикум
3	Лекция. Арифметические и логические операции. Приоритет операций	3	– применять арифметические операторы <code>+</code> , <code>-</code> , <code>*</code> , <code>/</code> , <code>//</code> , <code>%</code> , <code>**</code> ; строить логические выражения с <code>and</code> , <code>or</code> , <code>not</code> ; анализировать порядок вычислений	Фронтальный опрос
4	Практическая работа. Решение задач на вычисления и логику	3	– составлять программы для расчёта формул; отлаживать логические ошибки; тестировать код на различных входных данных	Самостоятельная работа
5	Лекция. Условные конструкции: <code>if</code> , <code>elif</code> , <code>else</code> . Вложенные условия	3	– писать программы с ветвлениями; использовать сравнения и логические связки; строить блок-схемы алгоритмов с условиями	Проверка кода
6	Практическая работа. Разработка программ с условными переходами	3	– решать задачи на классификацию данных; реализовывать меню-выбор; обрабатывать некорректный ввод пользователя	Мини-проект
7	Лекция. Циклы <code>while</code> и <code>for</code> . Операторы <code>break</code> , <code>continue</code>	3	– отличать циклы с предусловием и параметром; применять итерацию по последовательностям; управлять ходом выполнения цикла	Тестирование
8	Практическая работа. Программирование циклических алгоритмов	3	– реализовывать переборные задачи; считать суммы, произведения, количества;	Практикум

			отлаживать бесконечные циклы	
	Модуль 2. Структуры данных и функции	24		
9	Лекция. Строки: индексация, срезы, методы обработки	3	– работать с индексами и срезами строк; применять методы <code>.upper()</code> , <code>.split()</code> , <code>.replace()</code> ; форматировать вывод с <code>f-strings</code>	Проверка заданий
10	Практическая работа. Обработка текстовой информации	3	– анализировать и преобразовывать строки; искать подстроки, считать вхождения; генерировать отчёты в текстовом виде	Самостоятельная работа
11	Лекция. Списки и кортежи: создание, доступ, методы	3	– создавать и модифицировать списки; использовать <code>.append()</code> , <code>.pop()</code> , <code>.sort()</code> ; отличать изменяемые и неизменяемые типы	Фронтальный опрос
12	Практическая работа. Работа со списками: сортировка, фильтрация	3	– реализовывать алгоритмы поиска в списках; фильтровать данные по условиям; использовать списковые включения	Практикум
13	Лекция. Словари и множества: структура, применение	3	– создавать словари для хранения пар «ключ-значение»; применять множества для уникализации данных; выполнять операции объединения и пересечения	Проверка кода
14	Практическая работа. Разработка программ со словарями	3	– моделировать реальные объекты через словари; организовывать быстрый поиск данных; экспортировать/импортировать данные в JSON	Мини-проект
15	Лекция. Функции: определение, параметры, возврат значения	3	– выделять повторяющийся код в функции; использовать позиционные и именованные аргументы; документировать функции через <code>docstring</code>	Тестирование
16	Практическая работа. Модульное программирование	3	– разбивать задачу на подзадачи; импортировать стандартные модули <code>math</code> , <code>random</code> ; создавать собственные модули	Защита модуля
	Модуль 3. Основы электроники и микроконтроллеры	24		
17	Лекция. Основы электротехники: ток,	3	– изучать базовые электрические величины;	Входной контроль

	напряжение, сопротивление. Закон Ома		рассчитывать параметры цепей по закону Ома; читать простые принципиальные схемы	модуля
18	Практическая работа. Сборка простых цепей на макетной плате	3	– работать с breadboard, резисторами, светодиодами; измерять параметры мультиметром; соблюдать правила электробезопасности	Практикум
19	Лекция. Знакомство с микроконтроллером Arduino: архитектура, порты, питание	3	– изучать устройство и возможности Arduino; подключать плату к ПК, устанавливать драйверы; знакомиться со средой Arduino IDE	Инструктаж
20	Практическая работа. Первая программа для Arduino: мигание светодиодом	3	– писать скетчи на языке C++ для Arduino; компилировать и загружать код в плату; отлаживать аппаратно-программные ошибки	Практическая защита
21	Лекция. Цифровые входы/выходы. Кнопки, светодиоды, реле	3	– программировать чтение цифровых сигналов; управлять исполнительными устройствами; реализовывать защиту от дребезга контактов	Проверка скетча
22	Практическая работа. Интерактивные устройства: кнопка → светодиод	3	– собирать схемы с датчиками и индикаторами; писать программы реакции на внешние события; тестировать устойчивость работы системы	Мини-проект
23	Лекция. Аналоговые входы. Потенциометры, фоторезисторы, ШИМ	3	– изучать принцип АЦП и ШИМ-модуляции; считывать аналоговые значения с датчиков; плавно управлять яркостью/скоростью	Фронтальный опрос
24	Практическая работа. Регулировка параметров в реальном времени	3	– калибровать датчики под условия среды; реализовывать обратную связь в управлении; визуализировать данные через Serial Monitor	Практикум
	Модуль 4. Датчики, приводы и базовая робототехника	24		
25	Лекция. Ультразвуковые и инфракрасные датчики расстояния	3	– изучать физические принципы работы датчиков; рассчитывать погрешности измерений; выбирать датчик под задачу	Тестирование
26	Практическая работа. Измерение расстояния и	3	– подключать ультразвуковой датчик HC-	Практическая работа

	построение «карты» препятствий		SR04;обрабатывать эхо-сигналы в коде; фильтровать шумы и выбросы данных	
27	Лекция. Сервоприводы и шаговые двигатели: принцип, управление	3	– отличать типы приводов и области применения; рассчитывать угол/шаг по управляющему сигналу; использовать библиотеки `Servo`, `AccelStepper`	Проверка знаний
28	Практическая работа. Программирование движения механизмов	3	– управлять сервоприводом через потенциометр; реализовывать плавное позиционирование; калибровать механические ограничители	Мини-проект
29	Лекция. Датчики температуры, влажности, освещённости	3	– работать с датчиками DHT11/22, BH1750;интерпретировать данные согласно документации; учитывать условия эксплуатации сенсоров	Фронтальный опрос
30	Практическая работа. Сбор данных с датчиков окружающей среды	3	– считывать показания нескольких датчиков одновременно; усреднять данные для повышения точности; выводить результаты в консоль и на LCD-дисплей	Практикум
31	Лекция. Основы мобильной робототехники: шасси, колёса, баланс	3	– изучать кинематику колёсных платформ; рассчитывать передаточные отношения; планировать траектории движения	Проверка концепции
32	Практическая работа. Сборка и программирование робота-платформы	3	– монтировать механическую часть робота; подключать моторы и драйверы; писать программу прямолинейного движения	Защита модуля
	<b>Итого за 8 класс</b>	<b>96</b>		

## 9 КЛАСС (96 часов, 32 недели)

№ п/п	Наименование разделов и тем	Кол-во часов	Основные виды деятельности обучающихся	Формы контроля
	Модуль 5. Объектно-ориентированное программирование	24		

1	Лекция. Парадигмы программирования. Введение в ООП: классы, объекты	3	– сравнивать процедурный и объектный подходы; определять атрибуты и методы класса; создавать экземпляры классов	Входной контроль
2	Практическая работа. Разработка классов для моделирования объектов	3	– описывать классы с конструктором <code>__init__</code> ; реализовывать методы доступа к данным; тестировать объекты в интерактивном режиме	Практикум
3	Лекция. Инкапсуляция, свойства, приватные атрибуты	3	– скрывать внутреннюю реализацию через <code>__</code> и <code>__</code> ; использовать <code>@property</code> для геттеров/сеттеров; соблюдать принцип сокрытия деталей	Проверка кода
4	Практическая работа. Реализация инкапсуляции в проектах	3	– рефакторить процедурный код в ООП-стиль; защищать данные от некорректного изменения; документировать публичный интерфейс классов	Самостоятельная работа
5	Лекция. Наследование: переопределение методов, <code>super()</code>	3	– строить иерархии классов; расширять функционал базовых классов; использовать множественное наследование осознанно	Фронтальный опрос
6	Практическая работа. Разработка системы классов с наследованием	3	– создавать базовый класс «Датчик» и наследников; реализовывать полиморфизм через общий интерфейс; тестировать подстановку объектов	Мини-проект
7	Лекция. Абстрактные классы, интерфейсы, композиция	3	– применять модуль <code>abc</code> для абстракций; отличать наследование «является» от	Тестирование

			композиции «имеет»; проектировать гибкие архитектуры	
8	Практическая работа. Архитектура программы: от ТЗ до реализации	3	– декомпозировать задачу на классы и модули; применять паттерны «Стратегия», «Наблюдатель»; вести журнал изменений кода	Защита архитектуры
	Модуль 6. Продвинутая робототехника и управление	24		
9	Лекция. ПИД-регуляторы: теория и практика стабилизации	3	– изучать принципы пропорционально-интегрально-дифференциального регулирования; рассчитывать коэффициенты P, I, D; моделировать отклик системы	Проверка расчётов
10	Практическая работа. Реализация ПИД-регулятора для удержания линии	3	– программировать алгоритм следования по линии; калибровать коэффициенты под трассу; анализировать устойчивость движения	Практикум
11	Лекция. Обработка прерываний и таймеры в Arduino	3	– изучать аппаратные и программные прерывания; использовать <code>millis()</code> для неблокирующего кода; планировать задачи во времени	Фронтальный опрос
12	Практическая работа. Многозадачность в робототехнике	3	– реализовывать параллельное чтение датчиков и управление; избегать блокировок в цикле <code>loop()</code> ; отлаживать временные зависимости	Практическая работа
13	Лекция. Беспроводные интерфейсы: Bluetooth, Wi-Fi, NRF24L01	3	– сравнивать протоколы по дальности, скорости, энергопотреблению; настраивать модули HC-05, ESP-01; обеспечивать надёжность передачи	Проверка знаний

			данных	
14	Практическая работа. Управление роботом со смартфона через Bluetooth	3	– сопрягать модуль с мобильным устройством; разрабатывать протокол команд; реализовывать аварийную остановку	Мини-проект
15	Лекция. Протокол MQTT и облачные платформы для IoT	3	– изучать архитектуру «издатель-подписчик»; подключать устройства к брокеру MQTT; визуализировать данные	Тестирование
16	Практическая работа. Телеметрия робота в облако	3	– отправлять показания датчиков на сервер; строить графики в реальном времени; настраивать уведомления при превышении порогов	Защита модуля
	Модуль 7. Компьютерное зрение и элементы И	24		
17	Лекция. Введение в OpenCV: загрузка, отображение, базовые операции	3	– устанавливать библиотеку OpenCV-Python; читать и отображать изображения; выполнять преобразования цветового пространства	Входной контроль модуля
18	Практическая работа. Обработка изображений: фильтры, пороги, контуры	3	– применять размытие, детекцию краёв, бинаризацию; находить и выделять контуры объектов; рассчитывать геометрические характеристики	Практикум
19	Лекция. Распознавание цветов и маркеров для навигации	3	– калибровать цветовые диапазоны в HSV; детектировать цветные метки на изображении; вычислять координаты центра объекта	Проверка алгоритма
20	Практическая работа. Робот-слепопыт с компьютерным	3	– подключать веб-камеру к	Мини-проект

	зрением		одноплатному компьютеру; реализовывать алгоритм поиска и следования за меткой; оптимизировать производительность обработки	
21	Лекция. Введение в машинное обучение: классификация, регрессия	3	– отличать задачи обучения с учителем и без; готовить данные для обучения моделей; оценивать качество через метрики точности	Фронтальный опрос
22	Практическая работа. Классификация объектов с помощью scikit-learn	3	– обучать модель на размеченных данных; сохранять и загружать обученную модель; интегрировать предсказание в робототехнический проект	Практикум
23	Лекция. Нейросетевые подходы: свёрточные сети для изображений	3	– изучать архитектуру CNN на примере простых задач; использовать предобученные модели (MobileNet); адаптировать вывод под задачи робота	Тестирование
24	Практическая работа. Распознавание жестов для управления роботом	3	– собирать датасет жестов; дообучать модель под свои условия; реализовывать управление по распознанному жесту	Защита модуля
	Модуль 8. Проектная деятельность и подготовка к соревнованиям	24		
25	Лекция. Методология проектной работы: от идеи до прототипа	3	– формулировать проблему и цель проекта; составлять техническое задание и дорожную карту; оценивать риски и ресурсы	Проверка ТЗ
26	Практическая работа. Разработка концепции итогового проекта	3	– проводить мозговой штурм в команде; выбирать технологии под задачу;	Защита концепции

			распределять роли и этапы работы	
27	Лекция. Системы контроля версий: Git, GitHub для командной работы	3	– инициализировать репозиторий, делать коммиты; работать с ветками и пул-реквестами; разрешать конфликты слияния	Практикум по Git
28	Практическая работа. Коллективная разработка проекта в Git	3	– вести историю изменений кода и документации; организовывать code review; автоматизировать сборку через GitHub Actions	Проверка репозитория
29	Лекция. Подготовка к техническим соревнованиям: регламенты, критерии	3	– изучать требования Робофеста, WRO, НТИ; анализировать решения прошлых победителей; планировать стратегию выступления	Тестирование
30	Практическая работа. Тренировочные соревнования в классе	3	– выполнять задания на время и точность; отлаживать систему в условиях стресса; анализировать ошибки и улучшать решение	Соревновательный этап
31	Лекция. Презентация технических проектов: структура, визуализация, защита	3	– готовить слайды с акцентом на проблему и решение; демонстрировать работу устройства вживую; отвечать на вопросы жюри аргументированно	Репетиция защиты
32	Итоговая аттестация. Защита индивидуальных/групповых проектов	3	– продемонстрировать работоспособность проекта; представлять техническую документацию и код; рефлексировать результаты и планы развития	Экзамен / Защита
	<b>Итого за 9 класс</b>	<b>96</b>		

## РАЗДЕЛ 3. ФОРМЫ АТТЕСТАЦИИ

### Итоговая аттестация

Форма аттестации: Защита индивидуального или группового проекта (робототехнической системы или программного продукта) с демонстрацией работоспособности, представлением технического отчета и устным ответом на вопросы комиссии.

Критерии оценки: соответствие ТЗ, сложность алгоритма, надежность конструкции, качество кода, оригинальность решения, навыки презентации и командного взаимодействия.

## РАЗДЕЛ 4. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

### 4.1. Условия реализации программы

#### Материально-техническое обеспечение.

Помещение для проведения занятий должно соответствовать санитарным нормам и требованиям пожарной/электробезопасности. Каждый обучающийся имеет индивидуальное рабочее место (стол, стул, ПК/ноутбук). Рабочее место педагога оборудовано проектором/интерактивной панелью, принтером, маркерной доской.

Лаборатория оснащена: наборами микроконтроллеров (Arduino, ESP32, Micro:bit), комплектами датчиков и исполнительных механизмов, паяльным оборудованием, макетными платами, мультиметрами, осциллографами, конструкторами робототехнических платформ, 3D-принтером (при наличии). Обеспечен стабильный доступ к сети Интернет.

Информационное обеспечение: аудио-, видео- и презентационные материалы, электронные курсы, симуляторы (Tinkercad, Proteus, PyCharm Edu), раздаточные методические материалы, доступ к онлайн-библиотекам технической документации.

Кадровое обеспечение: педагог, имеющий высшее техническое или педагогическое образование (направление: информатика, робототехника,

электроника), подтвержденный опыт работы в сфере дополнительного образования или проектной деятельности с подростками, навыки программирования на Python/C++ и сборки робототехнических систем.

#### **4.2. Методическое обеспечение**

Основные методы организации занятий:

- словесные (лекция, беседа, технический брифинг, работа в парах/группах);
- практические (решение инженерных задач, лабораторные работы, сборка макетов, отладка кода, работа в группах);
- наглядные (показ видеоматериалов, демонстрация работающих прототипов, презентации, схемы);
- проектные и исследовательские (самостоятельное проектирование, прототипирование, экспериментальная проверка гипотез).

Методы, в основе которых лежит уровень деятельности обучающихся:

- репродуктивные (воспроизведение алгоритмов, следование инструкциям по сборке);
- частично-поисковые (участие в коллективном поиске оптимального архитектурного решения, модернизация готовых скетчей);
- исследовательские (самостоятельная разработка новых модулей, тестирование гипотез, участие в технических соревнованиях).

**Учебно-методический комплект включает:**

1. Дополнительная общеобразовательная программа.
2. Методические рекомендации для педагога (сценарии занятий, контрольные вопросы, критерии оценки).
3. Раздаточные материалы (блокноты инженера, чек-листы ТБ, инструкции по сборке, шаблоны технических отчетов).
4. Электронные ресурсы (закрытый репозиторий с кодом, видеозаписи разборов ошибок, база задач).