

Министерство просвещения Республики Башкортостан

Автономная некоммерческая организация развития системы подготовки кадров в сфере информационных технологий «Школа 21. Башкортостан»

Принята на заседании
Методического (педагогического) совета
от «19» 01 2026 года
протокол
№ 1 от «19» 01 2026 года

Утверждаю:
Директор АНО РСПК СИТ
«Школа 21. Башкортостан»
Е.С. Подымова
Приказ № 34-у от «19» 01 2026 года

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«L2. Программирование на Python, интернет вещей и основы машинного зрения»**

Направленность программы: техническая
Уровень освоения программы:
продолжающий
Возраст обучающихся: 12–14 лет
Срок реализации: 8 месяцев
Объем программы: 128 часов

Авторы-составители
эксперты АНО «Школа 21»

г. Уфа, 2026 год

ОГЛАВЛЕНИЕ

| | |
|--|-----------|
| РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ | 3 |
| 1.1. Пояснительная записка | 3 |
| 1.2. Цель и задачи программы | 7 |
| 1.3. Планируемые (ожидаемые) результаты программы | 8 |
| 1.4. Учебный план | 10 |
| 1.5. Содержание учебного плана | 12 |
| РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ | 15 |
| 2.1. Календарный учебный график | 15 |
| 2.2. Условия реализации программы | 15 |
| 2.3. Формы аттестации | 17 |
| 2.4. Оценочные материалы | 17 |
| 2.5. Материалы для промежуточной аттестации обучающихся по программе | 18 |
| 2.6. Материалы для итоговой аттестации обучающихся по программе | 26 |
| 2.7. Методические материалы | 33 |
| Методические материалы | 33 |
| Дидактические и лекционные материалы | 34 |
| 2.8. Список литературы | 35 |
| РАЗДЕЛ 3. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ | 37 |

РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

1.1. Пояснительная записка

Дополнительная общеобразовательная (общеразвивающая) программа «L2. Программирование на Python, интернет вещей и основы машинного зрения» предназначена для подростков в возрасте 12–14 лет, проявляющих интерес к современным цифровым технологиям, программированию, интернету вещей и машинному обучению.

Программа ориентирована на развитие у обучающихся базовых навыков программирования на языке Python, понимания архитектуры и принципов работы устройств на базе ESP32, а также первичного освоения технологий машинного зрения. Образовательный процесс строится на проектной деятельности и решении практико-ориентированных задач.

Образовательный процесс выстроен на практико-ориентированном подходе и включает в себя теоретическую часть, практические занятия и выполнение индивидуальных и групповых проектов. Программа предполагает использование современных микроконтроллеров (ESP32), языка программирования Python, а также базовых инструментов машинного обучения и машинного зрения.

Каждый обучающийся в процессе прохождения программы получает возможность пройти путь от простых программ до создания собственных «умных» устройств с элементами искусственного интеллекта, применяя полученные знания для решения реальных задач. Значительная часть программы отведена под работу с датчиками, камерами, облачными технологиями и алгоритмами, используемыми в современных IoT-системах и проектах машинного зрения.

Программа способствует формированию у школьников устойчивого интереса к инженерной деятельности и программированию, развитию творческого и логического мышления, а также пониманию того, как современные технологии используются для повышения качества жизни и решения социальных задач.

Направленность программы «L2. Программирование на Python, интернет вещей и основы машинного зрения» техническая.

Актуальность программы. Современный мир стремительно развивается в сторону цифровизации и автоматизации. Технологии интернета вещей, машинного обучения и искусственного интеллекта становятся неотъемлемой частью многих сфер жизни — от умного дома и медицины до транспорта, экологии и городской инфраструктуры. В этой связи востребованность специалистов, способных не только пользоваться цифровыми решениями, но и создавать их, продолжает стремительно расти.

Особую актуальность тема приобретает в контексте развития Национальной технологической инициативы (Постановление Правительства РФ от 18.04.2016 № 317 (ред. от 29.10.2024) «О реализации Национальной технологической инициативы»), программ «Цифровая экономика» (распоряжение Правительства Российской Федерации от 28 июля 2017 г. № 1632-р), а также задач, обозначенных в федеральных проектах, направленных на подготовку кадров для высокотехнологичных отраслей. Программирование, умные устройства и системы на базе машинного зрения входят в перечень ключевых технологических направлений будущего.

Программа «L2. Программирование на Python, интернет вещей и основы машинного зрения» позволяет обучающимся не просто знакомиться с современными цифровыми технологиями, а работать с ними «вживую» подключать датчики, создавать реальные устройства, взаимодействовать с микроконтроллерами и работать с компьютерным зрением. Это позволяет формировать у подростков представление о технологиях как о средстве улучшения качества жизни, в том числе через социально значимые проекты — например, устройство для слабовидящих или система автоматического полива.

Педагогическая целесообразность программы заключается в использовании современных и доказавших свою эффективность подходов проектного метода, метода «равный-равному», геймификации и цифровой платформы сопровождения. Чередование семинаров, кейсов, лабораторных и практических работ позволяет реализовать деятельностный подход, при котором учащиеся не просто получают теоретические знания, а сразу применяют их на практике, развивая инженерное мышление и технические навыки. Тематика программы отвечает интересам современных подростков и способствует профессиональной ориентации в сфере информационных технологий и инженерии. Занятия включают как индивидуальную, так и командную работу, что развивает навыки сотрудничества и ответственности. Таким образом, программа выполняет не только образовательную, но и развивающую, воспитательную функцию, обеспечивая всестороннее развитие личности учащегося.

Отличительные особенности программы.

Программа «L2. Программирование на Python, интернет вещей и основы машинного зрения» отличается междисциплинарным подходом, объединяющим программирование, интернет вещей и основы машинного обучения.

Учебный процесс строится на проектной деятельности, что позволяет применять знания на практике и повышает мотивацию обучающихся. Занятия направлены не только на освоение технических навыков, но и на развитие личностных качеств — инициативности, уверенности, ответственности. Используются современные инструменты платы ESP32, датчики, камеры и библиотеки машинного обучения. Программа ориентирована на решение реальных задач, включая социально значимые проекты. В рамках одной группы могут обучаться дети с разным стартовым уровнем (начальный, базовый, продвинутый). Обучающимся предоставляется возможность проявить индивидуальность, выбрать направление и уровень сложности проекта. Это делает программу актуальной, практико-ориентированной и значимой для профессионального самоопределения подростков.

Новизна программы. Новизна программы «L2. Программирование на Python, интернет вещей и основы машинного зрения» заключается в сочетании нескольких современных и востребованных направлений в одном курсе: программирование на Python, разработка IoT-устройств и начальное погружение в машинное обучение и компьютерное зрение.

Содержание программы разделено на два логически завершённых раздела:

– «Основы Python, IoT и веб-разработки» — формирует базовые навыки программирования и работы с «умными» устройствами.

– «Продвинутое программирование и машинное зрение» — развивает компетенции в области анализа данных и искусственного интеллекта.

Такое разделение обусловлено необходимостью последовательного усложнения материала: от работы с «железом» и базовым кодом к сложным алгоритмам обработки изображений. Объём каждого раздела составляет не менее восьми академических часов, что соответствует требованиям к минимальной насыщенности учебных блоков и позволяет глубоко проработать тему.

В отличие от традиционных курсов по программированию, данная программа ориентирована на создание полноценных умных устройств, взаимодействующих с физическим миром через сенсоры, камеры и микроконтроллеры. Применение технологий машинного зрения в проектах, таких как распознавание жестов или создание устройств для слабовидящих, делает содержание курса не только технически передовым, но и социально значимым. Интеграция проектной и личностно-ориентированной деятельности способствует формированию у подростков осознанного отношения к технологиям и их роли в жизни общества. Программа также предлагает уникальную возможность работы с реальными инженерными задачами в доступной форме, адаптированной под школьников. Всё это делает курс инновационным образовательным продуктом, направленным на

подготовку школьников к вызовам цифровой эпохи.

Ценность программы для ученика. Программа обладает уникальной практической ценностью, так как объединяет передовые образовательные методики с реальными задачами из сферы игровой индустрии, позволяя учащимся не просто изучать теорию, а сразу применять знания в профессиональном контексте. Использование P2P-механик создает среду, максимально приближенную к реальной рабочей атмосфере в индустрии информационных технологий, где важны коллаборация и обмен опытом. Технология коллаборативного обучения, лежащая в основе программы, развивает у учащихся критически важные навыки XXI века командную работу, коммуникацию, критическое мышление и способность к совместному решению сложных задач.

Особая ценность программы заключается в ее интерактивном формате через мозговые штурмы, групповые обсуждения и взаимную оценку работ учащиеся не только глубже усваивают материал, но и учатся конструктивно взаимодействовать, аргументировать свою точку зрения и принимать обратную связь. Такой подход превращает обучение в увлекательный процесс, где каждый участник становится активным создателем знаний, а не пассивным слушателем.

Уникальность программы.

| Особенность программы | В чем ее уникальность |
|--|--|
| Формат обучения «равный равному» | Развитие мягких навыков через методологию «равный равному», адаптированную под обучение детей. Обучение проходит на уникальной платформе школы, разработанной с учетом этой методологии. Дети учатся через взаимный анализ и поддержку, что формирует ответственность, умение объяснять решения и культуру «работы в команде как в реальной компании», доказавшую свою эффективность в Школе 21. |
| Игровая механика | «Внедрение игровых элементов способствует росту вовлеченности и интереса к учебе. Видя немедленный результат труда, учащиеся стремятся к новым вершинам. Механика включает личные и групповые задачи, накопление баллов, награды и единый сюжет, поощряя как индивидуальные, так и коллективные успехи. Процесс функционирует автоматически в среде платформы «Школы 21»» |
| Проектное обучение | Проектное обучение дает каждому участнику портфолио реальных задач такие проекты можно демонстрировать в вузах и на собеседованиях, подтверждая практическую компетентность выпускника. |
| Партнерство с ведущими компаниями информационных технологий | У Школы 21 большой пул отраслевых партнеров, за счет чего мы предоставляем проектные задачи на основе реальных бизнес-кейсов, а также устраиваем стажировки, экскурсии в IT-компании. Благодаря этому ученики получают прямой канал к будущему трудоустройству. |
| Гибкий подход к разработке и переработке программы на основе потребностей учеников | В процессе проведения программы мы регулярно собираем обратную связь с учащимися и используем продуктовый подход. На основе собранных данных программа лучше адаптируется под запросы детей и всегда отвечает запросам внешнего рынка. |

Образовательная деятельность в рамках программы направлена не только на освоение технических навыков, но и на:

- обеспечение духовно-нравственного, гражданско-патриотического воспитания обучающихся;
- формирование и развитие творческих способностей обучающихся;
- удовлетворение индивидуальных потребностей обучающихся в интеллектуальном, нравственном, художественно-эстетическом развитии и физическом совершенствовании;
- формирование культуры здорового и безопасного образа жизни, укрепление здоровья, а также на организацию свободного времени обучающихся;
- адаптацию обучающихся к жизни в обществе;
- профессиональную ориентацию обучающихся;
- выявление, развитие и поддержку талантливых обучающихся, проявивших выдающиеся способности;
- создание и обеспечение необходимых условий для личностного развития, профессионального самоопределения и творческого труда обучающихся;
- социализацию и адаптацию обучающихся к жизни в обществе; формирование общей культуры обучающихся.

Программа включает элементы, способствующие развитию ответственного отношения к физическому и эмоциональному состоянию учащихся, а также навыков безопасной работы с оборудованием и цифровыми технологиями.

В рамках реализации программы особое внимание уделяется следующим аспектам:

1. Формирование культуры безопасного труда и цифровой безопасности.

При работе с микроконтроллерами, датчиками, камерой ESP32-CAM и электротехническими компонентами ученики последовательно изучают правила безопасного обращения с оборудованием, правильной организации рабочего места, профилактики утомления и зрительного напряжения. На занятиях закрепляются навыки соблюдения техники безопасности и здорового цифрового поведения (режим работы за компьютером, цифровая гигиена).

2. Поддержка здоровой учебной среды.

Занятия строятся с чередованием теории, практики, активной деятельности и обсуждений в командах. Такой формат снижает уровень стресса, способствует осознанному распределению нагрузки и развивает навыки саморегуляции. Проектная работа требует планирования времени, что формирует у учащихся привычку контролировать ритм и объем деятельности.

3. Использование социально значимых проектов для укрепления ценности здорового образа жизни.

В программе предусмотрены проекты, связанные с мониторингом окружающей среды и качества жизни (умная метеостанция, системы автоматической сигнализации, IoT-устройства для контроля условий помещения). Это помогает подросткам осознать влияние среды на здоровье человека, а также увидеть, как технологии могут способствовать безопасности и благополучию.

4. Развитие навыков эмоционального благополучия.

Формат методологии «равный равному», применяемый на программе, предполагает взаимную поддержку, уважительное общение, конструктивную обратную связь и развитие командной культуры. Это способствует снижению тревожности, укреплению уверенности в себе, умению принимать ошибки как естественную часть обучения и формированию устойчивого «здорового» отношения к учебным сложностям.

5. Создание условий для активного и социально ориентированного досуга.

Проектные, исследовательские и творческие задания программы мотивируют подростков проводить свободное время за деятельностью, развивающей интеллект, креативность и практические навыки. Использование современных технологий в позитивном, созидательном контексте формирует ответственное отношение к

собственному развитию и здоровым формам досуга.

Адресат программы. Для обучения принимаются ученики общеобразовательных школ, обладающие базовыми знаниями физики, начальными навыками работы с компьютером.

Возраст обучающихся: 12-14 лет (6-8 класс)

Наполняемость группы: 8-12 человек

Объем программы 128 часов.

Срок освоения программы: 8 месяцев

Особенности организации образовательного процесса.

Форма обучения: очный формат обучения.

Форма организации образовательного процесса: групповые, индивидуально-групповые.

Режим занятий:

Продолжительность одного академического часа – 45 минут.

Перерыв между учебными занятиями – 10 минут.

Общее количество часов в неделю – 4 часа.

Занятия проводятся 2 раза в неделю по 2 часа.

Формы реализации образовательной программы. При проведении занятий будет использоваться индивидуально-групповая форма работы (Р2Р-механики, проектное обучение). В начале обучения будут преобладать занятия комбинированного типа, во второй половине – практические занятия.

Перечень форм проведения итогов реализации программы: тестирование.

1.2. Цель и задачи программы

Цель сформировать у подростков 12–14 лет практические навыки программирования, работы с IoT-устройствами и основами машинного зрения для решения реальных задач с использованием современных технологий.

Задачи программы

Обучающие

- обеспечить базовые знания по электронике, алгоритмизации и программированию на python, включая работу с переменными, циклами, условными операторами и функциями.
- обучить практическому использованию IoT-устройств подключению и настройке датчиков, сбору и обработке данных, передаче информации через интернет с применением платы esp32.
- научить создавать простые веб-интерфейсы для визуализации данных с датчиков и разработке пользовательских приложений с использованием flask.
- продемонстрировать применение знаний из физики, математики и информатики в рамках проектной деятельности.

Развивающие

- развивать познавательную активность, творческую инициативу и техническое мышление.
- формировать умение самостоятельно учиться, анализировать задачи, планировать и реализовывать проекты.
- развивать навыки работы с современными технологиями и методами обработки данных, включая основы машинного зрения.

Воспитательные

- воспитывать ответственность, самостоятельность, настойчивость в достижении результата.

- развивать коммуникативные качества и умение работать в команде, принимая участие в групповых проектах и обсуждениях.
- способствовать осознанному профессиональному самоопределению через знакомство с современными технологическими направлениями и создание социально ориентированных проектов.
- заинтересовать обучающихся дальнейшим углубленным изучением тематики в рамках следующей ступени обучения — программы I3 «машинное зрение»
- сформировать у обучающихся навыки безопасной работы с компьютерной техникой, включая соблюдение эргономических норм, гигиенических требований к организации рабочего места, режима зрительной нагрузки и рационального чередования труда и отдыха при работе с цифровыми устройствами.

1.3. Планируемые (ожидаемые) результаты программы

Метапредметные результаты

- развитие навыков самостоятельного обучения, умения анализировать технические задачи и планировать этапы реализации проекта от идеи до прототипа;
- сформированность умения работать в команде, распределять роли и задачи при разработке комплексных технических решений;
- развитие системного и алгоритмического мышления через проектирование решений, объединяющих программные и аппаратные компоненты;
- умение самостоятельно искать информацию в технической документации, справочных ресурсах и сообществах разработчиков для решения нестандартных задач;
- сформированность навыков презентации собственного проекта, умения аргументированно представить технические решения и выделить преимущества разработанного продукта;
- умение интерпретировать результаты работы системы, анализировать данные с датчиков и вносить обоснованные корректировки для улучшения функционала;
- развитие навыков критического мышления при оценке эффективности выбранных технических решений и инструментов.

Личностные результаты

- сформированность у обучающихся ответственного отношения к безопасной работе с компьютерной техникой и электрооборудованием, включая соблюдение эргономических норм и гигиенических требований;
- проявление осознанного интереса к сфере информационных технологий, интернету вещей и машинному зрению как к перспективным направлениям профессионального самоопределения;
- проявление настойчивости и целеустремлённости в достижении результата при отладке программ и сборке электронных устройств;
- сформированность представлений о ценности цифровых технологий и этических аспектах использования искусственного интеллекта и персональных данных;
- развитие познавательной активности и творческой инициативы в процессе создания социально-ориентированных технологических решений;
- сформированность уважительного отношения к мнению партнёров по команде и готовности к конструктивному диалогу в ходе совместной проектной деятельности.

Предметные результаты

- овладение базовыми конструкциями языка программирования Python, включая работу с переменными, условными операторами, циклами, функциями и основами ООП;
- умение настраивать среду разработки и писать программы на MicroPython для управления микроконтроллером ESP32;
- сформированность навыков подключения и программирования датчиков (температуры, влажности, расстояния, движения) для сбора и первичной обработки данных;
- овладение принципами передачи данных по протоколам I2C, SPI, UART и умение применять их при взаимодействии с периферийными устройствами;
- умение использовать структуры данных (списки, кортежи, словари) и формат JSON для организации и передачи информации в программах;
- сформированность навыков работы с веб-технологиями (HTML, CSS, базовый JavaScript) для создания интерфейсов визуализации данных с устройств;
- умение разворачивать веб-сервер на базе фреймворка Flask и организовывать беспроводной обмен данными между устройством и пользовательским интерфейсом;
- овладение базовыми понятиями машинного зрения и умение применять библиотеки OpenCV и NumPy для обработки изображений и работы с видеопотоком;
- умение использовать предобученные модели для детекции объектов (лица, руки, позы) и интеграции функций машинного зрения в комплексный проект;
- сформированность навыков тестирования IoT-систем, поиска и устранения ошибок в коде и аппаратной части устройства;
- умение применять знания из физики, математики и информатики в рамках комплексной проектной деятельности для решения прикладных задач;
- сформированность умения создавать простые веб-приложения для дистанционного мониторинга и управления устройствами на базе ESP32.

1.4.Учебный план

| № п / п | Наименование раздела/ темы | Количество часов | | | Формы аттестации / контроля |
|---|---|-------------------------|---------------------------|-----------------------|-----------------------------|
| | | Трудоемкость, теория | Трудоемкость, практика | Общая трудоемкость | |
| 1 | Вводное занятие. Знакомство с командой | 3 | 1 | 4 | - |
| Проект "Умный термометр" | | | | | |
| 2 | Знакомство с Python и платой ESP32 | 1 | 3 | 4 | ТК/оценка проектной работы |
| 3 | Логические операторы | 2 | 2 | 4 | ТК/оценка проектной работы |
| 4 | Условные операторы | 2 | 2 | 4 | ТК/оценка проектной работы |
| Проект "Метеостанция" | | | | | |
| 5 | Циклы | 2 | 2 | 4 | ТК/оценка проектной работы |
| 6 | Структуры данных языка Python | 2 | 2 | 4 | ТК/оценка проектной работы |
| 7 | Функции | 2 | 2 | 4 | ТК/оценка проектной работы |
| 8 | Знакомство с разметкой сайта и передача данных по WI-FI | 1 | 3 | 4 | ТК/оценка проектной работы |
| 9 | Знакомство с стилями. Создание сайта для метеостанции | 1 | 3 | 4 | ТК/оценка проектной работы |
| Групповой проект "Система автоматического полива растений" | | | | | |
| 10 | Знакомство с Flask, настройка сервера | 2 | 2 | 4 | ТК/оценка проектной работы |
| 11 | Кастомизация веб-страницы и сервера | 1 | 3 | 4 | ТК/оценка проектной работы |
| 12 | Командная работа | 0 | 4 | 4 | ТК/оценка проектной работы |
| 13 | Завершаем проект системы полива | 1 | 3 | 4 | ТК/ оценка проектной работы |
| Личный проект | | | | | |
| 14 | Концепция самостоятельного проекта | 1 | 3 | 4 | ТК/оценка проектной работы |
| 15 | Самостоятельная работа над проектом | 0 | 4 | 4 | ТК/оценка проектной работы |
| 16 | Дописываем приложение | 2 | 2 | 4 | ТК/презентация проектов |
| Проект по ООП: создание героев для RPG | | | | | |
| 17 | Знакомство с ИИ | 3 | 1 | 4 | ТК/оценка проектной работы |

| | | | | | |
|---|---|---|---|-----------------|----------------------------|
| 18 | Машинное зрение | 2 | 2 | 4 | ТК/оценка проектной работы |
| 19 | Проект по ООП | 2 | 2 | 4 | ТК/оценка проектной работы |
| Проект по машинному зрению для слабовидящих людей | | | | | |
| 20 | Знакомство с библиотекой OpenCV | 1 | 3 | 4 | ТК/оценка проектной работы |
| 21 | Создаем приложение | 1 | 3 | 4 | ТК/оценка проектной работы |
| 22 | Улучшаем приложение | 1 | 3 | 4 | ТК/оценка проектной работы |
| 23 | Тестируем приложение | 1 | 3 | 4 | ТК/оценка проектной работы |
| Проект по машинному зрению для распознавания жестов, перевода жестов в голос | | | | | |
| 24 | Знакомство с MediaPipe | 2 | 2 | 4 | ТК/оценка проектной работы |
| 25 | Сбор данных для машинного обучения | 2 | 2 | 4 | ТК/оценка проектной работы |
| 26 | Сбор данных и обучения модели | 1 | 3 | 4 | ТК/оценка проектной работы |
| 27 | Обучение модели | 1 | 3 | 4 | ТК/оценка проектной работы |
| 28 | Дорабатываем проект по распознаванию жестов | 0 | 4 | 4 | ТК/оценка проектной работы |
| Проект по созданию устройства умной сигнализации(фотоловушки) | | | | | |
| 29 | Начинаем проект умного устройства | 1 | 3 | 4 | ТК/оценка проектной работы |
| 30 | Сервер для проекта | 0 | 4 | 4 | ТК/оценка проектной работы |
| 31 | Тестируем приложение и исправляем ошибки | 0 | 4 | 4 | ТК/оценка проектной работы |
| 32 | Итоговое занятие. Презентация проектов, анонс L3 | 2 | 2 | 4 | ИА/итоговое тестирование |
| | | | | Итого 128 часов | |

1.5.Содержание учебного плана

Раздел 1. Основы Python, IoT и веб-разработки

Тема 1.1 Знакомство с командой

Теория. Знакомство с IoT и Python.

Практика. Установка среды разработки, изучение базового синтаксиса Python.

Тема 1.1 Знакомство с Python и платой ESP32

Теория. Основы работы с Python: ввод/вывод данных, переменные.

Практика. Работа с платой ESP32: подключение, прошивка, создание программы для управления светодиодом. Работа над проектом «Умный термометр».

Тема 1.2 Логические операторы

Теория. Изучение логических операторов.

Практика. Подключение цифровых датчиков к ESP32 и чтения их данных в терминал. Работа над проектом «Умный термометр».

Тема 1.3 Условные операторы

Теория. Работа с условными операторами (if, elif, else).

Практика. Подключение различных датчиков к ESP32 и программирования реакций на их показания. Работа над проектом «Умный термометр».

Тема 1.5 Циклы

Теория. Изучение циклов (while, for). Знакомство с протоколами передачи данных.

Практика. Применение протоколов при работе с датчиками на ESP32. Работа над проектом «IoT-метеостанция»

Тема 1.6 Структуры данных языка Python

Теория. Знакомство с форматом JSON и его ролью в вебе. Основы работы с API.

Практика. Освоение структур данных Python: списки, кортежи, словари.

Тема 1.7 Функции

Теория. Изучение функций: создание, вызов, область видимости.

Практика. Применение функционального подхода для структурирования кода на MicroPython.

Тема 1.8 Знакомство с разметкой сайта и передача данных по WI-FI

Теория. Основы веб-разметки (HTML).

Практика. Настройка ESP32 в качестве веб-сервера для передачи данных по Wi-Fi. Создание простой веб-страницы для отображения данных с датчиков.

Тема 1.9 Знакомство с стилями. Создание сайта для метеостанции

Теория. Стилизация веб-страниц с помощью CSS.

Практика. Динамическое обновление данных с датчиков ESP32 на веб-странице в реальном времени.

Тема 1.10 Знакомство с Flask, настройка сервера

Теория. Создание сервера на фреймворке Flask. Настройка виртуального окружения.

Практика. Организация беспроводной передачи данных между ESP32 и Flask-сервером. Работа над проектом «Система автоматического полива растений»

Тема 1.11 Кастомизация веб-страницы и сервера

Теория. Углубленная настройка Flask (view-функции, шаблонизатор).

Практика. Добавление базовой интерактивности на веб-страницу с помощью JavaScript. Адаптация сервера и сайта под задачи проекта.

Тема 1.12 Командная работа

Теория. Распределение ролей в команде.

Практика. Командная разработка проекта "Система автоматического полива": интеграция кода ESP32, сервера (Flask) и веб-интерфейса.

Тема 1.13 Завершаем проект системы полива

Теория. Интерпретация результатов работы системы.

Практика. Тестирование и отладка IoT-системы полива: проверка работы датчиков, сервера, веб-интерфейса.

Тема 1.14 Концепция самостоятельного проекта

Теория. Планирование индивидуального IoT-проекта. Определение функционала, выбор датчиков.

Практика. Проектирование взаимодействия между устройством (ESP32), сервером и веб-интерфейсом. «Работа над личным проектом»

Тема 1.15 Самостоятельная работа над проектом

Теория. Концепция индивидуального IoT-проекта.

Практика. Самостоятельная разработка: кодирование логики устройства (ESP32), настройка передачи данных.

Тема 1.16 Дописываем приложение

Теория. Анализ результатов модуля.

Практика. Доработка индивидуального проекта: добавление поддержки, отладка, оптимизация. Подготовка к презентации.

Раздел 2. Продвинутое программирование и машинное зрение

Тема 2.1 Знакомство с ИИ

Теория. Введение в искусственный интеллект. Основы ООП: классы, объекты, атрибуты, методы.

Практика. Создание классов для описания сущностей. Работа над проектом «Создание героев для RPG-игры»

Тема 2.2 Машинное зрение

Теория. Знакомство с машинным обучением и компьютерным зрением. Изучение наследования в ООП.

Практика. Создание иерархии классов.

Тема 2.3 Проект по ООП

Теория. Принципы практического применения ООП.

Практика. Проектирование и реализация классов для RPG-персонажей. Создание объектов, вызов методов, использование наследования.

Тема 2.4 Знакомство с библиотекой OpenCV

Теория. Освоение NumPy: создание и обработка массивов.

Практика. Работа с библиотекой OpenCV: захват видео, базовые операции с изображениями. Работа над проектом «Помощник для слабовидящих людей».

Тема 2.5 Создаем приложение

Теория. Принципы помощи слабовидящим с помощью CV.

Практика. Разработка приложения: обработка видеопотока с камеры в реальном времени с использованием OpenCV, наложение фильтров, сегментация изображения.

Тема 2.6 Улучшаем приложение

Теория. Принципы синтеза речи.

Практика. Улучшение приложения: добавление масок, более сложных фильтров обработки изображения. Интеграция модуля синтеза речи для озвучивания информации.

Тема 2.7 Тестируем приложение

Теория. Принципы тестирования приложений.

Практика. Тестирование и финальная доработка приложения для слабовидящих. Построение полного пайплайна обработки изображения и озвучивания результата.

Тема 2.8 Знакомство с MediaPipe

Теория. Знакомство с библиотекой MediaPipe.

Практика. Детекция лица, рук, позы человека в видеопотоке с использованием предобученных моделей. Работа над проектом «Переводчик жестов».

Тема 2.9 Сбор данных для машинного обучения

Теория. Подготовка данных для ML.

Практика. Сбор изображений жестов руками, разметка датасета для задачи распознавания жестов.

Тема 2.10 Сбор данных и обучения модели

Теория. Выбор алгоритма, оценка качества модели.

Практика. Подготовка данных, тренировка модели машинного обучения для распознавания жестов.

Тема 2.11 Обучение модели

Теория. Принципы интеграции моделей.

Практика. Использование модели для классификации жестов в реальном времени по видеопотоку. Озвучивание распознанных жестов.

Тема 2.12 Дорабатываем проект по распознаванию жестов

Теория. Методы оптимизации приложений.

Практика. Доработка проекта: добавление новых жестов, улучшение точности распознавания.

Тема 2.13 Начинаем проект умного устройства

Теория. Принципы работы ультразвуковых датчиков.

Практика. Подключение и настройка ультразвукового датчика к ESP32 CAM, программирование логики обнаружения движения. Работа над проектом «Камера с детекцией движения».

Тема 2.14 Сервер для проекта

Теория. Архитектура сервера для сигнализации.

Практика. Настройка сервера (Flask). Создание веб-интерфейса для отображения изображений с камеры и статуса системы.

Тема 2.15 Тестируем приложение и исправляем ошибки

Теория. Методы комплексного тестирования.

Практика. Проверка срабатывания датчика, захвата изображения, передачи данных на сервер, отображения в веб-интерфейсе. Поиск и исправление ошибок.

Тема 2.16 Презентация проектов, анонс L3

Теория. Анонс содержания и возможностей следующего уровня (L3).

Практика. Подготовка и проведение презентаций итоговых проектов. Обсуждение результатов модуля.

РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

2.1 Календарный учебный график

| № | Основные характеристики образовательного процесса | |
|----|---|----------------------|
| 1. | Количество учебных недель | 32 |
| 2. | Количество учебных дней | 64 |
| 3. | Количество часов в неделю | 4 |
| 4. | Количество часов | 128 |
| 5. | Неделя в 1 полугодии | 15 |
| 6. | Неделя во 2 полугодии | 19 |
| 7. | Начало занятий | 15 сентября |
| 8. | Каникулы | 31 декабря-11 января |
| 9. | Окончание учебного года | 31 мая |

2.2. Условия реализации программы

Материально-техническое обеспечение программы

Перечень оборудования, необходимых для занятий

| № | Наименование материалов | Количество |
|----|--|------------|
| 1. | Учебное помещение с посадочными местами на 12 учащихся | 1 |
| 2. | Персональные компьютеры с ОЗУ не менее 8 Гб, установленным пакетом Anaconda, доступом в интернет | 12 |
| 3. | Мультимедийный проектор и экран, или интерактивная доска | 1 |
| 4. | Плата ESP32 | 12 |
| 5. | Плата ESP32 CAM | 12 |
| 6. | Набор датчиков ARDUINO | 12 |

Характеристика помещений

| Наименование помещения / зоны | Материально-техническое оснащение |
|-------------------------------|---|
| Компьютерный класс | Помещение $\geq 2,5$ м ² на обучающегося; вентиляция; кондиционирование; освещенность ≥ 300 лк; напольное покрытие износостойкое; шумопоглощающие материалы; регулируемые столы; ортопедические стулья; стол преподавателя; ноутбуки 15.6", SSD 512 GB, RAM 16 GB; мыши; коврики; станции хранения ноутбуков; ИБП; Ethernet CAT6; Wi-Fi бесшовный; сетевые и электрические розетки; интерактивная доска 165 см. |
| Лаборатория электроники | Помещение ≥ 4 м ² на обучающегося; освещенность ≥ 500 лк; КЕО $\geq 1,5\%$; вентиляция; пожарная сигнализация; термостойкие столы; стулья; шкафы; мультиметры; ESP32, ESP32-CAM; макетные платы; датчики; инструменты; блоки питания; интерактивная доска; ноутбуки; соответствие нормам |

| | |
|-----------------------------|--|
| | оснащения кабинетов информатики. |
| Оснащение IoT / Python / CV | ESP32; ESP32-CAM; программаторы FTDI; датчики HC-SR04; платы IoT; кабели; Arduino IDE; PlatformIO; Python; Flask; OpenCV; Mediarpipe; MQTT-клиенты; оборудование для сервера. |
| Зона хранения оборудования | Металлические шкафы; станции хранения ноутбуков; контейнеры; стеллажи; маркировка; ограниченный доступ. |
| Лекционная зона | Мягкая мобильная мебель; $\geq 3,5$ м ² на человека; освещенность ≥ 300 лк; антивандальное покрытие; стеллажи; зарядные точки; медиаэкран/интерактивная доска. |
| Зона питания | Площадь ≥ 20 м ² ; столы; стулья; холодильник; микроволновая печь; кулер; шкафы; умывальник; вентиляция; освещенность ≥ 300 лк; контейнеры для мусора. |
| Гардероб | Площадь 15–25 м ² ; вешалки; пуфы; вентиляция; локеры; нескользящее покрытие; видеонаблюдение. |
| Санитарные комнаты | Отдельные санузлы; 8–10 м ² ; раковины; моющиеся поверхности; хранение инвентаря; доступность ОБЗ. |
| Ресепшн | Стойка; ПК; принтер; 4+ розеток; интернет; зона ожидания; освещенность ≥ 300 лк; домофон; видеонаблюдение. |
| Входная группа | Пандус; противоскользящие покрытия; освещённость ≥ 150 лк; камеры; планы эвакуации; пожарное оборудование; навигация. |
| Инженерные системы | Электроснабжение 2 категории; вентиляция; пожарная сигнализация; СКС CAT6; Wi-Fi; аварийное освещение; ИБП; видеонаблюдение. |

Кадровое обеспечение программы

Программа реализуется педагогом дополнительного образования (эксперт по сопровождению).

Требования к педагогу дополнительного образования:

Опыт работы:

1. 2+ года преподавания робототехники, электроники, мехатроники или смежных дисциплин подросткам 13–17 лет, включая проведение практических занятий с оборудованием и базовым программированием.
2. Владение ESP32, Arduino IDE, VS Code и навыки их настройки.
3. Опыт организации проектной деятельности или сопровождения групповых учебных проектов в формате «равный равному»

Образование:

1. Высшее или среднее техническое образование (робототехника, электроника, мехатроника, ИТ, инженерия).
2. Дополнительное педагогическое или психологическое образование / курсы повышения квалификации.

Профессиональные компетенции:

1. Владение KiCad и Blender (наличие портфолио с моделированием и проектами устройств).
2. Навыки работы с блочным программированием, Python, MQTT.
3. Базовое понимание машинного обучения, OpenCV, MediaPipe; умение работать с готовыми моделями и применять их на устройствах (TinyML).

Педагогические компетенции:

1. Навыки фасилитации и наставничества, ориентированные на самостоятельное обучение учеников.
2. Умение задавать наводящие вопросы и направлять к поиску решений.
3. Опыт работы с подростками и понимание их возрастных особенностей, мотивации и способов развития навыков самопроверки и обратной связи.

Личные качества:

1. Терпеливость, тактичность, умение поддерживать интерес к обучению.
2. Конструктивная коммуникация и выстраивание доверительных отношений.
3. Эмпатия, поддержка при ошибках, развитие мышления роста.
4. Гибкость и адаптивность в работе с группами разного уровня.

2.3. Формы аттестации

- Текущий контроль
- Промежуточная аттестация
- Итоговая аттестация

2.4. Оценочные материалы

| Этап контроля | Диагностическая методика | Средства контроля |
|--------------------------|---|--|
| Текущий контроль | Оценка проектной деятельности учащихся в ходе реализации мини-проектов | Лист наблюдения, чек-лист проектных умений, оценочные листы промежуточных результатов |
| Промежуточная аттестация | Защита итогового проекта с представлением функциональности и объяснением решений | Презентация проекта, защита, критерии оценивания по балльной системе (качество выполнения, креативность, обоснование решений). Тестирование. |
| Итоговая аттестация | Проверка углубленные знания, включая работу с функциями, классами, IoT-системами, серверными приложениями (Flask) и компьютерным зрением (MediaPipe). | Тестовые вопросы с выбором ответа (14 вопросов). Вопросы открытого типа. |

Средства контроля, позволяющие определить достижения планируемых результатов:

| Вид контроля | Цель | Методика | Система оценки | Интерпретация результата |
|---|--|--|---|---|
| Текущий контроль (оценка проектной работы) | оценить уровень сформированности практических навыков и усвоение учебного материала в процессе выполнения учебно-проектных заданий. Методика: | Наблюдение за выполнением практических заданий. Анализ промежуточных результатов проектной деятельности (мини-проектов, узловых заданий). Оценка | Каждое направление — до 5 баллов, всего — до 20 баллов. Контрольные листы и критерии позволяют объективно оценить текущие достижения | 18–20 баллов — высокий уровень; 14–17 — средний, необходима дополнительная проработка; <14 — начальный уровень, требуется повторное |

| | | | | |
|-----------------------------------|---|--|--|--|
| | | производится по критериям, разработанным под каждое направление. | | освоение материала. |
| Промежуточное тестирование | Проверить базовые знания и навыки учащихся по программированию на Python, работе с MicroPython, основам HTML, а также понимание принципов работы с датчиками и передачи данных. | Часть 1: Тестовые вопросы с выбором ответа (12 вопросов). Часть 2: Вопросы открытого типа, требующие написания кода, описания алгоритмов или объяснения концепций (8 вопросов). | Максимальный балл — 28 (12 за тестовую часть, 16 за открытые вопросы). | Уровни: слабый (0–13), средний (14–22), высокий (23–28). |
| Итоговая аттестация | Проверить углубленные знания, включая работу с функциями, классами, IoT-системами, серверными приложениями (Flask) и компьютерным зрением (MediaPipe). Формат: | Часть 1: Тестовые вопросы с выбором ответа (14 вопросов). Часть 2: Вопросы открытого типа, требующие написания кода, описания проектов и анализа технологий (9 вопросов). | Максимальный балл — 32 (14 за тестовую часть, 18 за открытые вопросы). | Уровни: слабый (0–15), средний (16–26), высокий (27–32). |

2.5. Материалы для текущего контроля обучающихся по программе

1. Проект "Умный термометр". Название на платформе - Smart_thermometer_L2

Цель проекта: создать IoT-устройство на базе ESP32, которое измеряет температуру с помощью датчика, анализирует данные и выводит уведомления ("холодно", "жарко") в зависимости от заданных условий.

Навыки, которые освоит ученик: работа с платой ESP32 и датчиками температуры; основы программирования на MicroPython (чтение данных, условия, циклы); подключение электронных компонентов (схемотехника); отладка и тестирование IoT-устройств.

Критерии оценивания: максимальный балл за проект: 15 баллов.

| Задание | Критерии (5 баллов) | Шкала |
|---------|---------------------|-------|
|---------|---------------------|-------|

| | | |
|-------------------------------------|---|--|
| Задание 1. Подключи датчик. | Корректность сборки схемы | 5 - Идеальное подключение, дополнительные улучшения 4 - Незначительные недочеты 3 - Есть проблемы в подключении 2 - Серьезные ошибки 1 - Схема нерабочая |
| Задание 2. Запрограммируй схему | Стабильность работы и качество кода | 5 - Безупречный код 4 - Мелкие недочеты 3 - Периодические сбои 2 - Код нестабилен 1 - Программа отсутствует |
| Задание 3. Добавь в схему словие | Корректность работы температурных условий | 5 - Расширенная логика с доп. реакциями 4 - Базовая реализация 3 - Частично рабочая 2 - Нестабильная работа 1 - Логика отсутствует |

2. Проект: «Метеостанция». Название на платформе: Weather_station_L2.

Цель проекта: создать IoT-устройство на базе ESP32, которое считывает данные с метеодатчиков (температура, влажность и др.), передаёт их на веб-страницу и отображает в реальном времени. Реализовать простую веб-визуализацию показаний.

Критерии оценивания: максимальный балл за проект: 20 баллов.

| Задание | Критерии (5 баллов) | Шкала |
|-------------------------------------|--------------------------------------|---|
| Задание 1. Подключи датчики. | Корректность сборки схемы. | 5 - Схема собрана идеально, аккуратное подключение всех датчиков, проверена полярность, надёжные соединения + добавлены дополнительные компоненты (например, конденсаторы, второй датчик). 4 - Схема собрана правильно, все датчики работают, подключение аккуратное, полярность соблюдена. 3 - Схема работает, но есть небольшие недочёты: неаккуратные провода, шаткие соединения, нет маркировки. 2 - Схема собрана, но есть ошибки в подключении (не тот пин, перепутана полярность, один датчик не работает). 1 - Схема отсутствует или собрана с грубейшими ошибками, компоненты не подключены, короткое замыкание, ничего не работает. |
| Задание 2. Запрограммируй схему. | Стабильность работы и качество кода. | 5 — Программа написана аккуратно, работает без ошибок, данные с нескольких датчиков стабильно выводятся в терминал, код читаемый + добавлены улучшения (например, преобразование единиц). 4 — Код работает, данные с датчиков приходят, программа корректна, но без дополнительных улучшений. 3 — Данные с датчиков поступают, но есть небольшие баги (например, нестабильный вывод, дублирование кода). 2 — Программа написана, но вывод некорректен или |

| | | |
|---|---|---|
| | | нестабилен, данные с датчиков искажаются. 1 — Программа не запускается, код отсутствует или сильно ошибочен. |
| Задание 3. Создай веб-страницу. | Корректность HTML-шаблона и стилей. | 5 — HTML-шаблон создан, применены стили, структура страницы чистая и логичная + добавлены элементы оформления (например, иконки, фон, блоки). 4 — Шаблон создан, стили применены, страница открывается и отображается корректно. 3 — Страница работает, но есть проблемы с версткой или стили минимальны. 2 — HTML-шаблон создан, но не открывается корректно, стили отсутствуют или не работают. 1 — Страница не создана, файл пустой или неработоспособный. |
| Задание 4. Реализуй динамическое отображение показаний датчиков на веб-странице. | Корректность обновления данных на веб-странице. | 5 — Показания с датчиков динамически обновляются на странице в реальном времени, передача реализована грамотно + оформление визуализации (например, график, цветовая индикация). 4 — Данные с датчиков передаются на страницу, обновляются корректно, но без визуального оформления. 3 — Передача данных есть, но обновление нерегулярное, задержка или проблемы с отображением. 2 — Попытка реализовать передачу данных есть, но страница не отображает их корректно. 1 — Динамическое отображение не реализовано, данные не передаются. |

3. Проект "Система автоматического полива растений". Название на платформе — `Smart_Garden_Helper_L2`

Цель проекта: создать IoT-устройство на базе ESP32, которое имитирует систему автоматического полива с возможностью удалённого управления через веб-интерфейс. Реализация будет выполнена с помощью светодиода, который символизирует включение и выключение насоса. Управление системой будет происходить через сервер, созданный с использованием фреймворка Flask.

Ключевые этапы

Задание 1: Создай схему.

Задание 2: Создай сервер с помощью Flask и реализуй отправку команд на плату ESP32.

Критерии оценивания: максимальный балл за проект: 10 баллов.

| Задание | Критерии (5 баллов) | Шкала |
|-----------------------------|--|---|
| Задание 1. Создай схему. | Создай схему — подключи ESP32 и светодиод, проверь полярность и соединения | 5 — Схема собрана идеально, аккуратно и надёжно. 4 — Всё работает, но без улучшений. 3 — Есть мелкие недочёты. 2 — Ошибки в подключении. 1 — Схема отсутствует или нерабочая. |
| Задание 2. | Создай сервер с | 5 — Сервер работает без ошибок, команды |

| | | |
|--|--|---|
| Создай сервер с помощью Flask и реализуй отправку команд на плату ESP32. | помощью Flask — реализуй управление «поливом» через веб-страницу | принимаются. 4 — Работает с мелкими ошибками. 3 — Страница есть, но управление не работает. 2 — Сервер создан, но страница не отображается. 1 — Сервер отсутствует. |
|--|--|---|

4. Личный проект. Название на платформе - **Personal_project_L2**

Цель проекта: создать полностью авторский проект — устройство, игру, веб-приложение или полезный гаджет — пройдя все этапы разработки: от идеи и планирования до технической реализации, дизайна и презентации. Проект должен продемонстрировать твои навыки и креативность, а также умение работать поэтапно.

Критерии оценивания: максимальный балл за проект: 20 баллов.

| Задание | Критерии (5 баллов) | Шкала |
|--|--|--|
| Задание 1. Планирование и идея | Планирование и идея — придумай концепт проекта, его цель и пользу. | 5 — Идея уникальна и хорошо продумана, есть чертежи/визуализация. 4 — Идея полезна и описана понятно, есть базовый план. 3 — Идея есть, но слабо проработана или не описана цель. 2 — Идея неясна, схема неполная. 1 — Идея отсутствует или скопирована. |
| Задание 2. Реализация технической части | Реализация технической части — собери схему или напиши код | 5 — Система полностью работает, аккуратная реализация. 4 — Работает, но есть мелкие баги. 3 — Работает частично, есть ошибки. 2 — Код/схема собраны, но не работают. 1 — Реализация отсутствует. |
| Задание 3. Дизайн/визуальное оформление | Дизайн и оформление — сделай проект удобным и красивым | 5 — Красивый, продуманный дизайн. 4 — Всё понятно и приятно выглядит. 3 — Работает, но оформление слабое. 2 — Нет стиля, сложно понять. 1 — Внешний вид отсутствует. |
| Задание 4. Презентация проекта | Презентация проекта — подготовь рассказ и покажи результат | 5 — Чёткая, уверенная презентация с демонстрацией. 4 — Презентация есть, немного сумбурная. 3 — Сложно понять суть проекта. 2 — Очень короткий рассказ, без демонстрации. 1 — Презентация не подготовлена. |

5. Проект "Создание героев для RPG-игры". Название на платформе - **RPG_hero_generator_L2**

Цель проекта: научиться работать с объектно-ориентированным программированием в Python: создавать классы с атрибутами, реализовывать методы (поведение объектов), создавать экземпляры классов, применять наследование для расширения возможностей.

Ключевые этапы:

1. Задание 1. Создай класс героя с заданными атрибутами
2. Задание 2. Создай методы класса

3. Задание 3. Создай несколько экземпляров класса
4. Задание 4. С помощью наследования создай дочерний класс

Критерии оценивания: максимальный балл за проект: 20 баллов.

| Задание | Критерии (5 баллов) | Шкала |
|---|---------------------------------------|--|
| Задание 1. Создай класс героя с заданными атрибутами | Создай класс Hero с атрибутами | 5 — Все атрибуты на месте и работают. 4 — Есть 1–2 недостающих атрибута. 3 — Мелкие недочёты, неинформативные имена. 2 — Нет атрибутов. 1 — Класс не создан или нерабочий. |
| Задание 2. Создай методы класса | Создай методы класса | 5 — Все методы работают логично. 4 — Не хватает 1–2 методов. 3 — Методы есть, но с недочётами. 2 — Реализованы частично или с ошибками. 1 — Методы отсутствуют. |
| Задание 3. Создай несколько экземпляров класса | Создай несколько экземпляров класса | 5 — Герои разные по характеристикам и ролям. 4 — Один экземпляр, всё работает. 3 — Различия минимальны. 2 — Повторяющиеся характеристики. 1 — Экземпляры не созданы. |
| Задание 4. С помощью наследования создай дочерний класс | Создай дочерний класс с наследованием | 5 — Добавлены уникальные атрибуты/методы. 4 — Есть наследование и новое свойство/метод. 3 — Отличия минимальны. 2 — Ошибки в наследовании. 1 — Наследование отсутствует. |

6. Проект по машинному зрению для слабовидящих людей. Название на платформе - AI_vision_assistant_L2.

Цель проекта: разработать приложение на базе компьютерного зрения (OpenCV), которое: детектирует и сегментирует объекты в реальном времени; озвучивает названия предметов для помощи слабовидящим людям.

Ключевые этапы:

Задание 1. Настраиваем модель на детектирование предметов с помощью сегментации

Задание 2. Добавляем озвучивание предметов

Критерии оценивания: максимальный балл за проект: 10 баллов.

| Задание | Критерии (5 баллов) | Шкала |
|---|--|---|
| Задание 1. Настраиваем модель на детектирование предметов с помощью | Развернуть библиотеку OpenCV. Реализовать детекцию и | 5 — Модель успешно развёрнута, сегментация реализована корректно и стабильно, код работает без ошибок, кадр обрабатывается в реальном времени 4 — Сегментация работает, OpenCV установлен, |

| | | |
|--|--|--|
| сегментации | сегментацию объектов в реальном времени. | код выполняется корректно, результат адекватный, но есть мелкие недочеты в коде 3 — Сегментация реализована, но есть недочёты: кадр обновляется с задержкой, цветовая маска неочевидна, не все объекты корректно обрабатываются. 2 — Библиотека установлена, но код с ошибками, сегментация работает частично или нестабильно. 1 — Модель не работает, код отсутствует или выдаёт ошибки, сегментация не реализована. |
| Задание 2. Добавляем озвучивание предметов | Интегрировать модуль синтеза речи. Настроить проговаривание названий обнаруженных объектов. | 5 — Озвучивание реализовано чётко и синхронно, программа озвучивает названия предметов в реальном времени, голос настроен 4 — Озвучивание работает корректно, названия объектов проговариваются, но есть мелкие недочеты в коде 3 — Озвучка работает, но со сбоями: звук накладывается, запаздывает или повторяется. 2 — Попытка озвучивания есть, но звук воспроизводится неправильно или не соответствует предметам/не производится совсем. 1 — Озвучивание не реализовано или вызывает ошибки, модуль не подключен. |

7. Проект "Язык жестов в речь". Название на платформе - Gesture_to_voice_converter_L2

Цель проекта: разработать приложение на базе библиотеки MediaPipe, которое: детектирует жесты рук в реальном времени; распознает их с помощью модели KNN; озвучивает жесты синтезатором речи для помощи людям с нарушениями слуха/речи.

Ключевые этапы:

Задание 1. Развертывание модели, настройка базового детектирования

Задание 2. Сбор данных

Задание 3. Обучение модели KNN

Задача 4. Добавление озвучивания жестов в программу

Критерии оценивания: максимальный балл за проект: 20 баллов.

| Задание | Критерии (5 баллов) | Шкала |
|--|---|--|
| Задание 1. Развертывание модели, настройка базового детектирования | MediaPipe установлен и работает идеально, детекция рук работает стабильно, код аккуратный | 5 – MediaPipe установлен и работает идеально, детекция рук работает стабильно, код аккуратный 4 – Библиотека установлена, руки детектируются корректно, есть мелкие недочеты в коде 3 – Детекция работает, но с небольшими недочетами (например, не хватает отображения, не все случаи распознаются корректно). 2 – Детекция настроена частично, возникают ошибки или отображение нестабильно. 1 – Детекция не работает, код с ошибками, библиотека не используется по назначению. |
| Задание 2. Сбор | Реализован | 5 – Реализован собственный сборщик данных, |

| | | |
|---|--|--|
| данных | собственный сборщик данных, выбрано несколько жестов, всё структурировано и сохранено корректно. | выбрано несколько жестов, всё структурировано и сохранено корректно. 4 – Собраны данные по каждому жесту,, но они одинаковые, без изменения положения жеста 3 – Данные собраны, но не по всем жестам или в меньшем объеме, структура частично хаотична. 2 – Данных мало или они плохо структурированы, есть ошибки при сборе. 1 – Сбор данных не реализован или выполнен с критическими ошибками. |
| Задание 3. Обучение модели KNN | Все данные объединены, модель обучена и протестирована, работает стабильно | 5 – Все данные объединены, модель обучена и протестирована, работает стабильно 4 – Модель обучена на корректных данных, есть редкие небольшие ошибки в результатах 3 – Модель обучена, но работает неуверенно или с ошибками, классификация жестов слабая. 2 – Модель не обучена корректно, данные не подготовлены, есть ошибки в коде. 1 – Модель не обучена, код отсутствует или не запускается. |
| Задача 4. Добавление озвучивания жестов в программу | Озвучивание подключено, работает в реальном времени, каждое распознанное движение вызывает корректную реплику, + добавлены разные фразы или языки. | 5 – Озвучивание подключено, работает в реальном времени, каждое распознанное движение вызывает корректную реплику, + добавлены разные фразы или языки. 4 – Звук работает стабильно, каждый жест озвучивается правильно, интеграция выполнена хорошо. 3 – Озвучивание добавлено, но есть задержки, ошибки воспроизведения или повторов. 2 – Озвучка реализована частично или с существенными багами. 1 – Озвучивания нет, модуль не подключен, или голос не работает. |

8. Проект "Захват кадра при детекции движения". Название на платформе - Motion_capture_camera_L2

Цель проекта: создать систему на базе esp32-cam, которая: детектирует движение с помощью датчика/камеры; автоматически захватывает и сохраняет кадры; передает изображения на веб-сервер (Flask) для просмотра и анализа.

Ключевые этапы:

1. Задание 1. Программирование платы
2. Задание 2. Сервер на Flask
3. Задание 3. Настройка взаимодействия сервера и платы

Критерии оценивания: максимальный балл за проект: 15 баллов.

| Задание | Критерии (5 баллов) | Шкала |
|-----------------------------------|-------------------------------------|--|
| Задание 1. Программирование платы | Программа полностью рабочая. Датчик | 5 – Программа полностью рабочая. Датчик подключён корректно, данные считываются точно и стабильно. |

| | | |
|--|--|---|
| | <p>подключён корректно, данные считываются точно и стабильно.</p> | <p>4 – Код работает, но данные с датчиков поступают с перебоями или некорректно интерпретируются.</p> <p>3 – Программа выполняет основную функцию, но с ошибками (неправильная логика, задержки не настроены).</p> <p>2 – Код написан, но работает неправильно.</p> <p>1 – Код отсутствует или не запускается, программа не компилируется.</p> |
| <p>Задание 2. Сервер на Flask</p> | <p>Сервер стабильно работает, веб-страница отображается корректно, функции управления работают без ошибок, код структурирован.</p> | <p>5 – Сервер стабильно работает, веб-страница отображается корректно, функции управления работают без ошибок, код структурирован.</p> <p>4 – Сервер работает, веб-страница отображается, управление работает, но есть мелкие баги или нестабильность</p> <p>3 – Сервер работает и открывает веб-страницу, но взаимодействие с платой не реализовано или не функционирует</p> <p>2 – Сервер запускается, но страница не отображается или отображается неправильно (например, ошибки CSS, нет подключения). Управление отсутствует.</p> <p>1 – Сервер не создан. Flask не установлен, сервер не создаётся или не запускается</p> |
| <p>Задание 3. Настройка взаимодействия сервера и платы</p> | <p>Отправка кадра реализована безупречно: изображение поступает, обрабатывается и отображается на сервере. Всё работает автоматически, без ошибок.</p> | <p>5 – Отправка кадра реализована безупречно: изображение поступает, обрабатывается и отображается на сервере. Всё работает автоматически, без ошибок.</p> <p>4 – Кадр приходит на сервер, но есть мелкие недочеты в логике (например, периодическое подвисание, ошибки отображения, некорректная реакция сервера на файл).</p> <p>3 – Кадр поступает, но наблюдаются проблемы с датчиком (размытость, ошибки при съемке) или нестабильная передача (пропуски, задержки).</p> <p>2 – Кадр передаётся, но часто возникают сбои: программа зависает, сервер перестает принимать данные, логика рушится при повторных попытках.</p> <p>1 – Передача не реализована. Кадр не приходит на сервер. Сервер не получает данные от платы или возникает критическая ошибка.</p> |

2.6. Материалы для итоговой аттестации обучающихся по программе

1. Финальный проект "Проект 8. Камера с детекцией движения"

Проект направлен на создание интеллектуальной системы наблюдения на базе ESP32-CAM. Устройство фиксирует движение с помощью ультразвукового датчика, делает снимок и передает его на веб-сервер, где фото отображается на веб-странице.

Проект включает элементы электроники, программирования микроконтроллеров, веб-разработки и основ компьютерного зрения.

Шаги выполнения проекта

| Задание | Критерии (5 баллов) | Шкала |
|--|--|---|
| <p>Шаг 1. Подготовка и настройка оборудования</p> <p>Подключить ультразвуковой датчик HC-SR04 к ESP32-CAM.</p> <p>Настроить прошивку и подключение платы к Wi-Fi.</p> <p>Проверить работу камеры и получение данных с датчика.</p> | <p>Настройка ESP32-CAM и логика детекции движения (1–5 баллов)</p> <p>Что оценивается:</p> <ul style="list-style-type: none"> корректность подключения датчика; стабильность считывания данных; работа логики фиксации движения; успешное подключение к Wi-Fi. | <p>Критерии:</p> <ul style="list-style-type: none"> 5 — схема собрана идеально, данные стабильны, код полностью рабочий; 4 — работа корректная, но с небольшими перебоями; 3 — основная логика есть, но присутствуют ошибки; 2 — код работает некорректно; 1 — код отсутствует или не запускается. |
| <p>Шаг 2. Программирование ESP32-CAM</p> <p>Написать код, который:</p> <ul style="list-style-type: none"> подключается к сети Wi-Fi; измеряет расстояние до объекта; фиксирует движение при расстоянии меньше порога; делает снимок; отправляет фото на сервер через HTTP POST. | <p>Создание веб-сервера и веб-страницы (1–5 баллов)</p> <p>Что оценивается:</p> <ul style="list-style-type: none"> корректность запуска Flask-сервера; работа маршрутов (endpoints); отображение страницы в браузере; структура кода и наличие HTML-шаблона. | <p>Критерии:</p> <ul style="list-style-type: none"> 5 — сервер стабилен, страница отображается правильно, код структурирован; 4 — сервер работает, но есть мелкие недочеты; 3 — страница открывается, но взаимодействие с платой не реализовано; 2 — сервер запускается, но страница не отображается корректно; 1 — сервер отсутствует/не запускается. |
| <p>Шаг 3. Создание веб-сервера на Flask</p> <p>Установить Flask и подготовить минимальный сервер.</p> <p>Реализовать приём файлов (фотографий) от платы.</p> <p>Настроить сохранение поступивших изображений.</p> <p>Создать HTML-страницу для отображения последнего полученного кадра.</p> | <p>Интеграция устройства и сервера (1–5 баллов)</p> <p>Что оценивается:</p> <ul style="list-style-type: none"> корректность передачи данных между ESP32-CAM и Flask; качество | <p>Критерии:</p> <ul style="list-style-type: none"> 5 — фото приходит автоматически, отображается корректно, система работает стабильно; 4 — фото приходит, но есть небольшие задержки или баги; 3 — снимок приходит, но качество/стабильность низкие; |
| <p>Шаг 4. Интеграция всех компонентов</p> <p>Настроить отправку снимков с ESP32-CAM на Flask-сервер.</p> <p>Проверить, что сервер корректно принимает фото и обновляет страницу.</p> <p>Тестировать систему: движение</p> | <p>Критерии:</p> <ul style="list-style-type: none"> 5 — фото приходит автоматически, отображается корректно, система работает стабильно; 4 — фото приходит, но есть небольшие задержки или баги; 3 — снимок приходит, но качество/стабильность низкие; | <p>Критерии:</p> <ul style="list-style-type: none"> 5 — фото приходит автоматически, отображается корректно, система работает стабильно; 4 — фото приходит, но есть небольшие задержки или баги; 3 — снимок приходит, но качество/стабильность низкие; |

| | | |
|---|--|--|
| <p>→ снимок → отображение на сайте.</p> | <p>полученного снимка; стабильность обновления страницы; отсутствие ошибок передачи.</p> | <p>2 — попытка интеграции есть, но система работает нестабильно; 1 — фото не поступает, интеграция не выполнена.</p> |
| <p>Шаг 5. Финальная сборка и демонстрация Сфотографировать финальную схему устройства. Сделать короткое видео-демо работы системы. Подготовить итоговые файлы: код сервера, код платы, HTML-страницы.</p> | | |

Перечень вопросов итогового тестирования:

Часть 1: Тестовые вопросы

1. Что делает `def` в Python?

- Запускает условие
- Объявляет функцию (верный ответ)
- Завершает программу
- Создает переменную

2. Какой результат вернет функция?

```
def sum(a, b):  
    return a + b  
print(sum(3, 5))
```

- 8 (верный ответ)
- 35
- 3+5
- Ошибка

3. Для чего используется библиотека `machine` в MicroPython?

- Для создания сайтов
- Для управления пинами и модулями платы (верный ответ)
- Для рисования
- Для генерации случайных чисел

4. Какой из форматов подходит для обмена данными между ESP32 и веб-сервером?

- .docx
- .json (верный ответ)
- .jpg
- .exe

5. Как подключить датчик к ESP32?

- Через интернет
- Через пины GND, VCC, и data (верный ответ)
- Через PowerPoint
- С помощью Wi-Fi

6. Какой цикл подходит для бесконечного выполнения кода?

- `for i in range(5)`
- `if True`
- `while True` (верный ответ)
- `while False`

7. Что делает этот код?

```
if temp > 30:  
    fan.on()
```

- Всегда включает вентилятор
- Включает вентилятор, если температура выше 30 (верный ответ)
- Отключает вентилятор
- Ошибка

8. Что делает Flask?

- Управляет микроконтроллером
- Создает серверное приложение (верный ответ)
- Открывает веб-камеру
- Подключает датчики

9. Какая структура данных позволяет хранить пару "ключ: значение"?

- список
- словарь (верный ответ)
- кортеж

- цикл
- 10. Что делает MediaPipe?**
- Управляет светодиодами
 - Распознает лицо, руки, позу в видео (верный ответ)
 - Сохраняет файлы
 - Создает таблицы
- 11. Как передать данные с ESP32 на веб-страницу?**
- Через CSS
 - Через Wi-Fi и HTTP-запрос (верный ответ)
 - Через Bluetooth
 - Через Python Turtle
- 12. Что такое ООП в Python?**
- Один общий параметр
 - Объектно-ориентированное программирование (верный ответ)
 - Операционная оптическая программа
 - Обмен оптическими переменными
- 13. Что обозначает class?**

`class Robot:`

`pass`

- Это цикл
 - Это список
 - Это определение шаблона объекта (верный ответ)
 - Это переменная
- 14. Что такое метод `__init__` в классе?**
- Функция старта, запускаемая при создании объекта (верный ответ)
 - Условие
 - Переменная
 - Стил HTML

Часть 2: Вопросы открытого типа

15. Напиши свою простую функцию на Python, которая включает полив, если влажность меньше 40.

(Можно условие и команду включения, псевдокод)

Пример правильного ответа:

```
def check_moisture(moisture):
```

```
    if moisture < 40:
```

```
        print("Включить полив")
```

```
    else:
```

```
        print("Полив не нужен")
```

16. Опиши шаги, чтобы с нуля создать IoT-систему «умная метеостанция», которая показывает температуру на сайте.

(Можно кратко: датчик → ESP → сервер → сайт)

Примерный правильный ответ:

- Подключить датчик температуры к плате ESP32.
- Написать код на MicroPython для чтения данных с датчика.
- Настроить Wi-Fi-соединение на ESP32.
- Отправлять данные на веб-сервер (например, через HTTP-запрос).
- Создать HTML-страницу, которая отображает полученные данные.

17. Ты хочешь, чтобы твоя система отправляла уведомление, если температура выше нормы.

Как это можно реализовать?

Примерный правильный ответ:

- В коде на ESP32 использовать условие `if temp > порог`:
- Если условие выполнено, отправлять HTTP-запрос на сервер
- Также можно использовать свет или звук в устройстве как сигнал тревоги.

18. Опиши, чем отличается переменная от функции в Python.

Приведи свой пример.

Примерный правильный ответ:

- Переменная — это "ящик", в котором храним данные. Например: `temp = 25`
- Функция — это "действие", набор команд, который можно вызывать.

19. В проекте на ESP32 тебе нужно запускать код, пока не получены нужные данные.

Какой цикл ты используешь и почему?

Примерный правильный ответ:

- Использую `while` или `while True`:
- Цикл будет повторяться, пока условие (например, `данные == None`) выполняется
- Это позволяет устройству постоянно ждать или слушать входящий сигнал

20. Представь, что ты делаешь систему для слабовидящих людей. Что бы ты распознавал с помощью камеры и как это улучшит жизнь?

Примерный правильный ответ:

- Камера может распознавать лица, надписи, дорожные знаки, жесты
- Система может проговаривать вслух, что видит, с помощью голосового модуля
- Это поможет человеку ориентироваться в пространстве и получать важную информацию

21. В чём польза P2P-проверки для тебя лично? Что ты понял, проверяя чужие проекты?

Примерный правильный ответ:

- Я увидел, как по-разному можно решить одну задачу
- Когда проверяешь другого, начинаешь лучше понимать свой код
- Иногда замечаешь идеи, которые можно использовать у себя
- Это как работать в команде — и учиться, и помогать

22. Что тебе больше всего понравилось в курсе? Что бы ты добавил или изменил?

23. Опиши свой финальный проект (или идею). Что он делает и какие технологии ты использовал?

Критерии оценивания итогового тестирования

Часть 1: Тестовые вопросы

| № | Вопрос | Правильный ответ | Балл |
|---|---|---|------|
| 1 | Что делает <code>def</code> в Python? | Объявляет функцию | 1 |
| 2 | Что вернёт <code>sum(3, 5)</code> | 8 | 1 |
| 3 | Зачем нужна <code>machine</code> в <code>MicroPython</code> | Для управления пинами | 1 |
| 4 | Формат для передачи данных | .json | 1 |
| 5 | Как подключить датчик | Через пины GND, VCC и data | 1 |
| 6 | Цикл для бесконечного выполнения | <code>while True</code> | 1 |
| 7 | Что делает <code>if temp > 30: fan.on()</code> | Включает вентилятор, если температура выше 30 | 1 |
| 8 | Что делает Flask | Создает серверное приложение | 1 |

| | | | |
|----|-----------------------------|---|---|
| 9 | Структура “ключ: значение” | словарь | 1 |
| 10 | Что делает MediaPipe | Распознает лицо, руки, позу в видео | 1 |
| 11 | Как передать данные на сайт | Через Wi-Fi и HTTP-запрос | 1 |
| 12 | Что такое ООП | Объектно-ориентированное программирование | 1 |
| 13 | Что делает class | Определяет шаблон объекта | 1 |
| 14 | Что такое __init__ | Функция запуска при создании объекта | 1 |

Максимум за часть 1: 14 баллов

Часть 2: Вопросы открытого типа

| № | Вопрос | Критерии (по 2 балла за каждый) | Макс. балл |
|----|-------------------------------|---|------------|
| 15 | Функция полива | есть if, есть условие < 40, print | 2 |
| 16 | Этапы IoT-системы | шаги в правильной последовательности (датчик, ESP, Wi-Fi, сервер, сайт) | 2 |
| 17 | Система уведомлений | условие + способ уведомления (сайт, звук, свет) | 2 |
| 18 | Переменная и функция | корректное сравнение, свой пример | 2 |
| 19 | Цикл для ожидания | выбор while + объяснение зачем | 2 |
| 20 | Идея системы для слабовидящих | что распознаёт + как помогает | 2 |
| 21 | R2P-проверка | минимум 2 адекватных вывода/преимущества | 2 |
| 22 | Отзыв о курсе | аргументированное мнение (что понравилось и/или что можно улучшить) | 2 |
| 23 | Финальный проект | цель проекта + технологии (ESP, датчики, Flask и т.д.) | 2 |

Максимум за часть 2: 18 баллов

Итоговые баллы

| Уровень | Баллы (из 32) | Интерпретация |
|---------|---------------|--|
| Слабый | 0–15 | Базовые знания, нужна помощь с проектной логикой |
| Средний | 16–26 | Уверенно ориентируется, готов к проектам и команде |
| Высокий | 27–32 | Отлично освоил темы, способен к самостоятельной разработке |

Критерии оценки успешного освоения программы

Для успешной аттестации

1. обучающийся посетил не менее 26 занятий в рамках программы.
2. все предусмотренные учебным планом проекты выполнены в полном объеме и соответствуют установленным требованиям.
3. обучающийся подготовил и провел презентацию итогового проекта, продемонстрировав приобретённые знания и навыки.
4. итоговый тест по программе пройден с результатом не ниже 60%.
5. обязательное участие обучающегося в групповом проекте, включая активное взаимодействие с командой и выполнение своей части работы.

2.7. Методические материалы

Методические материалы

Лабораторные/практические работы:

- Изучение базового синтаксиса Python: переменные, условные операторы, циклы, структуры данных, функции.
- Подключение платы ESP32 к компьютеру, настройка прошивки MicroPython.
- Сбор данных с датчиков (температура, влажность, освещенность) и передача на ПК.
- Разработка веб-страницы для отображения данных, подключение CSS для стилизации.
- Создание простого сервера на Flask для приёма и отправки данных.
- Основы ООП: создание классов, объектов, методов и наследования.
- Проектирование игровых объектов на Python (пример — герои RPG).
- Подключение и настройка камеры ESP32-CAM.
- Работа с OpenCV: чтение изображений, базовая обработка, распознавание лиц и объектов.
- Разработка алгоритмов распознавания жестов и преобразования их в голос.
- Создание системы «умной сигнализации» с функцией автоматической фотофиксации.

Методика:

- Демонстрация и разбор кода с пошаговым объяснением.
- Парная работа: один участник пишет код, другой проверяет и отлаживает.
- Постепенное усложнение задач: от простого вывода данных до интеграции в веб-интерфейс.
- Мини-проекты на каждом этапе: «Умный термометр», «Метеостанция».
- Кейсы, приближенные к реальным задачам (распознавание лиц, охранные системы, помощь слабовидящим).
- Исследовательский подход: тестирование разных алгоритмов и параметров обработки.
- Групповая работа над крупными проектами и индивидуальное доведение идеи до прототипа.

Дидактические и лекционные материалы

Лекции/презентации:

- Основы Python и ООП.
- Архитектура IoT-систем, подключение и программирование ESP32.
- Основы веб-разработки: HTML, CSS, Flask.
- Введение в машинное зрение и OpenCV.
- Интеграция искусственного интеллекта в IoT-проекты.

Рекомендуемые инструменты и оборудование:

- **ПО:** Python 3.x, Thonny, Jupyter Notebook, Arduino IDE, Flask, OpenCV.
- **Оборудование:** ПК, ESP32, ESP32-CAM, датчики, макетные платы, провода.
- **Материалы:** презентации, схемы подключения, примеры кода, ссылки на GitHub/Google Drive с проектами.

Методики исследовательской и опытнической работы

Форматы:

- Мини-исследования в рамках лабораторных работ (например: «Как частота опроса датчиков влияет на скорость обновления данных на сайте»).
- Проектно-исследовательская деятельность: создание прототипа IoT-системы или

устройства машинного зрения.

- Постановка гипотезы, проведение эксперимента, анализ результатов, оформление отчета и презентации.

Методические принципы:

- Обучение через исследование (*inquiry-based learning*).
- Принцип «peer-to-peer» — обучение «равный-равному» с элементами геймификации (очки опыта, уровни).
- Пошаговое проектирование с обязательной промежуточной проверкой работоспособности на каждом этапе.

Тематика опытнических/исследовательских работ:

1. Сравнение точности разных датчиков температуры и влажности.
2. Влияние задержек в коде на стабильность работы IoT-устройства.
3. Распознавание лиц при разных условиях освещения.
4. Эффективность фильтров OpenCV для выделения объектов на фото.
5. Создание системы автоматического реагирования на события (сигнализация, оповещения).

Основная концепция обучения на курсе – «равный-равному». В основе концепции лежит идея сотрудничества в сочетании с самостоятельным поиском решений с применением открытых образовательных ресурсов и материалов, разработанных авторами программы и размещенных на образовательной платформе. Также, данная модель обучения дополнена элементами геймификации получение очков опыта (XP) и достижение уровней. Различные элементы геймификации мотивируют учеников выполнять задания и достигать учебных целей.

2.8. Список литературы

Основные источники

1. Бейтс, Б. «Программирование на Python для начинающих». — М. Диалектика, 2019. — 320 с.
2. Гэрретт, К. «Введение в машинное обучение». — СПб. Питер, 2020. — 400 с.
3. Джонс, М. «Основы Интернета вещей (IoT)». — СПб. Питер, 2021. — 256 с.
4. Иванов, А. П. «Машинное зрение теория и практика». — М. Горячая линия — Телеком, 2020. — 400 с.
5. Кнут, Д. Э. «Искусство программирования». — СПб. Питер, 2017. — 672 с.
6. Ларсон, Р. «Введение в микроконтроллеры ESP32». — М. БХВ-Петербург, 2022. — 288 с.
7. Митчелл, Т. «Машинное обучение». — М. Вильямс, 2018. — 512 с.
8. Смирнова, Е. В. «Основы веб-разработки для IoT». — СПб. Питер, 2021. — 350 с.
9. Чен, Ю. «Python для Интернета вещей». — М. БХВ-Петербург, 2021. — 400 с.

Дополнительные источники

1. Альберт, К. «Практическое программирование на Python». — М. БХВ-Петербург, 2020. — 350 с.
2. Браун, Д. «Интернет вещей архитектура и приложения». — СПб. Питер, 2022. — 280 с.
3. Гусев, И. В. «Введение в машинное зрение и обработку изображений». — М. Горячая линия — Телеком, 2019. — 320 с.
4. Джеймс, Г., Уиттен, Д., Хасты, Т., Тибширани, Р. «Введение в статистическое обучение». — М. Вильямс, 2016. — 400 с.
5. МакКинни, У. «Python и машинное обучение». — СПб. Питер, 2021. — 450 с.
6. О'Рейли, Т. «Основы Flask создание веб-приложений». — М. БХВ-Петербург, 2020. — 300 с.
7. Ривз, М. «Программирование ESP32 от новичка к профессионалу». — М. Диалектика, 2021. — 360 с.
8. Сингх, А. «Компьютерное зрение для начинающих». — СПб. Питер, 2020. — 280 с.

Интернет-ресурсы:

1. Python и основы программирования

1. <https://www.python.org> — официальный сайт языка Python (документация, загрузка, примеры).
2. <https://docs.python.org/3/tutorial/index.html> — официальный учебник Python.
3. <https://realpython.com> — статьи, проекты и уроки по Python.
4. <https://pypi.org> — каталог библиотек Python.

2. IoT и ESP32

1. <https://randomnerdtutorials.com> — подробные проекты с ESP32, ESP8266, датчиками и камерой.
2. <https://micropython.org> — официальный сайт MicroPython.
3. <https://docs.espressif.com> — документация и SDK для ESP32 от производителя Espressif.

3. Веб-разработка (HTML, CSS, Flask)

1. <https://developer.mozilla.org> — документация и учебники по HTML, CSS, JavaScript.
2. <https://flask.palletsprojects.com> — официальный сайт Flask.
3. <https://jinja.palletsprojects.com> — шаблонизатор Jinja2 (используется в Flask).

4. Машинное зрение и OpenCV

1. <https://opencv.org> — официальный сайт OpenCV.
2. <https://docs.opencv.org/master> — документация OpenCV.
3. <https://github.com/opencv/opencv> — исходный код и примеры.

5. Образовательные платформы

1. <https://stepik.org> — бесплатные курсы по Python, IoT, машинному обучению.
2. <https://github.com> — хостинг для кода и проектов.
3. <https://colab.research.google.com> — бесплатная среда для запуска Python и машинного зрения в браузере.

6. Дополнительные ресурсы

1. <https://wokwi.com> — онлайн-симулятор Arduino и ESP32.
2. <https://circuitpython.org> — CircuitPython для IoT-устройств.
3. <https://www.tinkercad.com> — онлайн-среда для моделирования схем и 3D-печати.

РАЗДЕЛ 3. ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Алгоритм — последовательность действий или инструкций, предназначенных для решения задачи или выполнения определенной функции.

API (Application Programming Interface) — интерфейс программирования приложений, набор инструментов и протоколов для создания программного обеспечения.

Датчик — устройство, которое преобразует физический параметр (температуру, влажность и др.) в электрический сигнал для дальнейшей обработки.

ESP32 — микроконтроллер с поддержкой Wi-Fi и Bluetooth, широко используемый для создания IoT-устройств.

Интернет вещей (IoT) — сеть физических устройств, оснащенных программным обеспечением и сенсорами, которые взаимодействуют и обмениваются данными через интернет.

Машинное обучение — область искусственного интеллекта, в которой компьютерные системы обучаются распознавать закономерности и принимать решения на основе данных.

Машинное зрение — технология, позволяющая компьютерам анализировать и интерпретировать визуальную информацию из изображений или видео.

Переменная — именованная область памяти, используемая для хранения данных в программе.

Python — высокоуровневый язык программирования, известный своей простотой и удобством для обучения.

Сервер — компьютер или программа, предоставляющая сервисы и ресурсы другим компьютерам в сети.

Функция — именованный блок кода, выполняющий определенное действие и возвращающий результат.

Цикл — конструкция в программировании, позволяющая повторять набор команд несколько раз до выполнения условия.