

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ АДМИНИСТРАЦИИ ЗАТО Г. СЕВЕРОМОРСК

Муниципальное бюджетное учреждение дополнительного образования
ЗАТО г. Североморск «Дом детского творчества им. Саши Ковалева»

Программа принята
методическим советом
МБУДО «ДДТ»
Протокол №3
«21» апреля 2025г.

УТВЕРЖДАЮ
Директор МБУДО «ДДТ»
 М.Б.Клыкова
«13» мая 2025г.
Приказ №160

Дополнительная
общеразвивающая программа технической направленности

«От Scratch к Arduino: ОСНОВЫ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ И ЭЛЕКТРОНИКИ»

Срок реализации 1 год
для учащихся от 7-11 лет

Автор:

педагог

дополнительного образования

Кривцова

Елена Васильевна

ДОКУМЕНТ ПОДПИСАН
УСИЛЕННОЙ КВАЛИФИЦИРОВАННОЙ
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 45 b6 9c 04 7e 17 78 da 5c 98 97 9c e2 af 10 c1
Владелец: Клыкова Марина Байраковна
Действителен: с 28.04.2025 до 22.07.2026

г. Североморск
2025

Сегодня наука и технология развиваются столь стремительно, что образование зачастую не успевает за ними. Одним из самых известных неформальных способов организации внеучебной образовательной деятельности является метод проектов. Самым подходящим инструментом для организации такой деятельности является среда Scratch, которая есть серьезное и современное направление компьютерного дизайна и анимации. Овладев даже минимальным набором операций, самый неискушённый пользователь может создавать законченные проекты.

Метод знакомства с робототехникой через творческие и технические проекты позволяет не только выявить из большого числа учащихся самых увлеченных и работоспособных, но и создать необходимые условия и мотивацию для овладения каждого ребенка методологией творческой деятельности.

Пояснительная записка

При составлении образовательной программы были учтены требования к современной образовательной программе, которые изложены в Федеральном законе «Об образовании в Российской Федерации» от 29.12.2012 N 273-ФЗ, Приказе Министерства просвещения РФ от 27 июля 2022г. N629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам», Постановлением Главного государственного санитарного врача Российской Федерации от 28.09.2020 г. № 28 «Об утверждении санитарных правил СП 2.4. 3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления учащихся и молодежи», Постановлением Главного государственного санитарного врача Российской Федерации от 28 января 2021 года № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания», Распоряжением Правительства Мурманской области от 03.07.2019 № 157-РП «О концепции внедрения целевой модели региональной системы дополнительного образования учащихся в Мурманской области», Постановлением администрации ЗАТО г. Североморск № 556 от 01.04.2020 г. «Об утверждении Положения о персонифицированном дополнительном образовании в ЗАТО г. Североморск», Уставом учреждения.

Программа предназначена для учащихся 8-11 лет. Срок реализации программы - 8 месяцев. Уровень освоения программы – стартовый.

Актуальность программы

Актуальность программы обусловлена развитием современных и перспективных технологий. Данная образовательная программа позволяет не только обучить созданию программ, но и подготовить обучающихся к планированию и организации работы над разноуровневыми техническими проектами. Это в дальнейшем поможет осуществить осознанный выбор вида деятельности в техническом творчестве.

Робототехника является популярным и эффективным методом для изучения важных областей науки, технологии, конструирования и физики. Доступность микроконтроллеров, удобные среды для программирования, выбор образовательных конструкторов дают возможность реализоваться даже не самым технически заинтересованным детям. Обучение робототехнике способствует ранней профориентации, успешной реализации будущих инженеров особенно в метапредметной области, на стыке дисциплин.

Новизна программы

Аспект новизны заключается в том, что Scratch не просто язык программирования, а еще и интерактивная среда, где результаты действий визуализированы, что делает работу с программой понятной, интересной и увлекательной. Отличительной особенностью программы является ее адаптивность как для обучающихся не имеющих ранее первичного навыка работы в области программирования так, и для обучающихся, имеющих опыт работы в иных средах. На занятиях по программе осуществляется работа с образовательными

конструкторами на платформе Arduino, программирование осуществляется в программе на базе Scratch.

В обучении по данной программе используются игровые технологии. В играх у обучающихся вырабатываются стратегии жизненного поведения. В строительстве моделей закрепляются навыки технологических приёмов. При отработке неудач прочно усваиваются законы физики, а при поиске решения открытой задачи используются знания из других наук.

Педагогическая целесообразность

Педагогическая целесообразность данной программы обусловлена необходимостью подготовить обучающихся к изучению в будущем языков программирования и заложить у них хороший фундамент в развитии логического мышления, а также сформировать навыки к работе с электроникой. Педагогический потенциал среды программирования Scratch позволяет рассматривать ее как перспективный инструмент и средство организации межпредметной проектной познавательной деятельности учащегося, направленной на его личностное и творческое развитие. Scratch выступает в качестве инструмента создания разнообразных творческих проектов: мультфильмов, игр, рекламных роликов, музыки, «живых» рисунков, интерактивных историй и презентаций, компьютерных моделей, обучающих программ для решения образовательных задач: обработки и отображения данных, закрепления и коррекции умений и навыков, моделирования, управления устройствами и развлечения. Во время занятий обучающиеся научатся проектировать, создавать и программировать на базе Arduino. При обучении по программе закладываются основы исследовательской работы и проектного мышления при реализации собственных идей.

Практическое программирование в данном курсе проходит в игровой, увлекательной форме. При создании практикума особое внимание уделяется структурированию материала, содержательному исполнению каждой работы. Задания подобраны так, чтобы выработать у учащихся устойчивые навыки работы, программируя вначале простые проекты, постепенно усложняя их.

Цель программы - развитие интеллектуальных и творческих способностей учащихся средствами изучения и создания визуальных сред для программирования, формирование начальных технических компетенций в области робототехники, развитие критического мышления и творческого потенциала через освоение робототехнической платформы «Arduino».

Достижение поставленной цели предполагает решение ряда **задач**:

Обучающие:

- познакомить с основами программирования;
- сформировать основы информационной и алгоритмической культуры;
- развитие навыка разработки проектов: интерактивных историй, игр, обучающих программ, мультфильмов, моделей, свободного проектирования и др.;
- ознакомление с принципами сборки схем на макетных платах;
- формирование навыков чтения простых схем;
- формирование навыков работы с датчиками, сенсорами, светодиодами, дисплеями, двигателями, совместимыми с платформой Arduino;
- ознакомление со средой программирования Arduino IDE;
- создание условий для профессионального самоопределения, творческой самореализации.

Развивающие:

- развитие конструкторских навыков;
- развитие эстетического сознания через творческую деятельность на базе иллюстрированной среды программы Scratch;
- развитие интереса к научно-техническому, инженерно-конструкторскому творчеству.
- развитие логического мышления и творческих способностей;

- развитие пространственного воображения;
- формирование и развитие информационной компетенции: навыков работы с различными источниками информации, умения самостоятельно искать, извлекать и отбирать необходимую для решения учебных задач информацию;
- развитие фантазии и воображения и творческих способностей;
- развитие изобретательских, конструкторских способностей;
- развитие самостоятельности и формирование умения работать в паре, малой группе, коллективе.
- расширить кругозор учащихся в области информационных технологий.

Воспитательные:

- формирование мотивации к изобретательству;
- воспитание стремления к получению качественного законченного материала;
- воспитание умения работать в группе;
- формирование нравственных качеств личности и культуры поведения в обществе.

Исходя из поставленных задач, важнейшими принципами построения программы являются:

- научность и доступность: использование на занятиях доступных для детей понятий и терминов, учет уровня подготовки, опора на имеющийся у учащихся опыт от простого к сложному;
- системность, последовательность и доступность в освоении трудовых приемов: изучение нового материала опирается на ранее приобретенные знания;
- гуманистический характер отношений педагога и ребенка: ребенок рассматривается как активный субъект совместной с педагогом деятельности, основанной на реальном сотрудничестве, уважении к личности и демократическом стиле взаимоотношений педагога с детьми;
- принцип природосообразности - предполагает учет возрастных и индивидуальных особенностей, задатков, возможностей детей при включении их в различные виды деятельности;
- поощрение творческих достижений учащихся, самостоятельности при выполнении творческих практических работ.

Программа представляет собой логически выстроенную систему, направленную, с одной стороны, на овладение знаниями в интересующей учащегося области, с другой стороны, ориентированную на формирование у ребенка целостной научно-технической картины мира. Материал программы систематизирован, доступно и логично излагается, подкреплён мощным дидактическим материалом, направлен на практику основ программирования, на развитие творчества и самостоятельности учащихся общего развития, исходя из диагностики их возможностей. Возможно обучение по индивидуальной образовательной траектории, включение в состав одной учебной группы обучающихся с разным уровнем подготовки и прохождение обучения по индивидуальному образовательному маршруту.

Обучение строится таким образом, чтобы учащиеся хорошо усвоили приемы работы в среде программирования, научились «читать и понимать» простейшие алгоритмы и программы, а затем собирать простые схемы. Содержание нацелено на активизацию познавательной деятельности каждого обучающегося с учетом его возрастных особенностей. Значительно внимание уделяется повышению мотивации.

Количество детей в группе 12 человек. Группа делится на два звена. Распределение учебной нагрузки представлено следующим образом: ((1 раз в неделю по 2 часа) x 32 недели) x 2 звена=128 часов или ((2 раза в неделю по 1 часу) x 32 недели) x 2 звена = 128 часов;

Один академический час занятия в группе равен 45 минутам астрономического времени.

В объединение принимаются все желающие дети с 8-11 лет по заявлению родителей

или лица их заменяющего. При формировании групп учитывается возраст учащихся и их подготовка в области компьютерной грамотности.

Формы, методы и режим занятий

Основная форма занятий - практикум. Предполагает знакомство учащихся с основами физики, в частности с понятием электрического тока, сопротивлением, силой тока, последовательным и параллельным соединением, законом Ома, а также с техникой безопасности при составлении схем электрических цепей, также учащиеся знакомятся с составными элементами вычислительной платформы Arduino и компонентами.

Кроме того, на занятиях применяются следующие формы и методы обучения:

- лекция – предназначена для изучения несложного, но большого объема теоретического материала; теоретических основ по каждой теме;
- учебная дискуссия – стимулирование познавательного интереса, вовлечение учащихся в активное обсуждение разных научных точек зрения по той или иной проблеме, побуждение их к осмыслению различных подходов к аргументации чужой и своей позиции. Учит учащихся мыслить, спорить, доказывать свою правоту.
- лекция с элементами беседы – более продуктивный метод по сравнению с предыдущим, за счёт общего разбора с учащимися наиболее сложных и важных вопросов в каждой теме;
- моделирование информационного процесса, ситуации – воспитанникам предлагается реальная жизненная ситуация для оценки её с точки зрения информатики и информационных технологий;
- групповое задание – воспитанники объединяются в группы и разбирают предлагаемую им задачу коллективно. После этого происходит обсуждение со всей группой, выявление и анализ допущенных ошибок;
- проектная деятельность - создание и защита собственного или группового проекта (творческого задания) позволяет наиболее широко раскрыть умственный и творческий потенциал воспитанников, научиться работать в коллективе;
- игра – ролевые, деловые, дидактические, развивающие компьютерные игры, тренажеры;
- наглядные методы используются во взаимосвязи со словесными и практическими методами обучения и предназначены для наглядно-чувственного ознакомления учащихся с явлениями, процессами, объектами в их натуральном виде или в символическом изображении с помощью всевозможных рисунков, репродукций, схем и т. п.;
- объяснительно-иллюстративные (демонстрация готовых иллюстраций, работа по таблицам и схемам);
- частично-поисковый (выполнение вариативных заданий);
- проблемно-поисковый (в проектной деятельности);
- обучающий контроль;
- творческие (творческие задания, проекты).

Для оценки эффективности образовательной программы разработан оценочно-результативный блок (Приложение 1):

Весь материал курса сгруппирован в четыре раздела:

Вводное занятие. Правила техники безопасности при работе в компьютерном классе;

Раздел 1. «Проекты в Scratch»

Раздел 2. «Эксперименты с блоками»

Раздел 3. «Основные понятия электроники»

Раздел 4. «Проектная деятельность»

Обучение начинается со знакомства детей с основами работы с интерфейсом программы Scratch. На данном этапе происходит формирование творческого коллектива.

Затем происходит более глубокое и детальное знакомство учащихся с принципами программирования Arduino, способами и приемами соединения деталей и далее с интерфейсом и основными командами среды программирования.

Учащиеся выполняют проекты самостоятельно.

Учебно-тематический план обучения

№ п/п	Тема	Количество часов			Формы аттестации
		Теори я	Практи ка	Всего	
Раздел 1. «Проекты в Scratch»					
1.1	Вводное занятие. Инструктаж по технике безопасности в кабинете информатики и робототехники. Основы работы на компьютере. Основы программирования в среде Scratch и интерфейс программы.	3(1,5)	1(0,5)	4(2)	Опрос
1.2	Опыты с растровой графикой и векторной графикой.	-	4(2)	4(2)	Практическое задание
1.3	Циклы. Опыты с блоками Движения.	-	4(2)	4(2)	Практическое задание
1.4	Начальная расстановка спрайтов. Опыты с блоками Внешность.	1(0,5)	3(1,5)	4(2)	Опрос
1.5	События. Проект «Начинающий музыкант»	-	2(1)	2(1)	Практическое задание
1.6	Опыты с числами. Координаты.	1(0,5)	1(0,5)	2(1)	Практическое задание
1.7	Опыты со словами. Вращение и градусы.	2(1)	2(1)	4(2)	Практическое задание
Раздел 2. Эксперименты с блоками					
2.1	Сообщения. Эксперимент «Фрукты в корзину», Эксперимент «Волшебное зелье».	1(0,5)	3(1,5)	4(2)	Практическое задание
2.2	Условия и операторы выбора. Эксперимент «Выключаем звезды»	1(0,5)	1(0,5)	2(1)	Опрос
2.3	Изменение координат. Процедуры. Эксперимент «Лабиринт»	1(0,5)	5(2,5)	6(3)	Практическое задание
2.4	Логические операторы. Цикл с условием. Эксперимент «Кнопка лифта».	1(0,5)	1(0,5)	2(1)	Практическое задание
2.5	Случайные числа и диапазоны. Эксперимент «Дождь».	1(0,5)	1(0,5)	2(1)	Практическое задание
2.6	Эксперимент «Чат-бот».	-	4(2)	4(2)	Опрос
2.7	Переменные в играх. Эксперимент «Счет».	-	4(2)	4(2)	Практическое задание
Раздел 3. Основные понятия электроники					
3.1	Знакомство с контроллером Ардуино.	1(0,5)	3(1,5)	4(2)	Опрос
3.2	Основы проектирования и моделирования электронного устройства на базе Ардуино.	1(0,5)	3(1,5)	4(2)	Практическое задание
3.3	Широтно-импульсная модуляция.	2(1)	2(1)	4(2)	Практическое задание
3.4	Программирование Ардуино. Пользовательские функции.	2(1)	2(1)	4(2)	Опрос
3.5	Сенсоры. Датчики Ардуино.	2(1)	2(1)	4(2)	Практическое задание
3.6	Кнопка – датчик нажатия.	1(0,5)	3(1,5)	4(2)	Практическое задание
3.7	Цифровые индикаторы. Семисегментный индикатор.	1(0,5)	3(1,5)	4(2)	Практическое задание

3.8	Жидкокристаллический экран.	2(1)	4(2)	6(3)	Практическое задание
Раздел 4. Проектная деятельность					
4.1	Кнопочный переключатель	-	6(3)	6(3)	Практическое задание
4.2	Светильник с кнопочным Управлением	-	4(2)	4(2)	Практическое задание
4.3	Кнопочные ковбои	-	4(2)	4(2)	Практическое задание
4.4	Счётчик нажатий	-	4(2)	4(2)	Практическое задание
4.5	Комнатный термометр	-	6(3)	6(3)	Практическое задание
4.6	Пантограф	-	6(3)	6(3)	Практическое задание
4.7	Перетягивание каната	-	8(4)	8(4)	Практическое задание
4.8	Создание проекта.	-	8(4)	8(4)	Конкурс
	Итого:	24(12)	104(52)	128(64)	

() – время работы одного звена

Содержание программы

Вводное занятие. Правила техники безопасности при работе в компьютерном классе. Основы работы на компьютере.

Правила техники безопасности при работе в компьютерном классе. Правила работы за компьютером (включение и выключение компьютера).
(Презентация 1.)

Раздел 1. Проекты в Scratch

1.1. Основы программирования в среде Scratch.

Теория. Среда программирования Scratch: история создания, области применения. Основные определения: «алгоритм», «программа», «команда», «система команд исполнителя», «исполнитель», «сцена», «проект», «спрайт», «скрипт». (Презентация 2.)

Практика. Знакомство с программой Scratch: интерфейс программы, просмотр команд разных блоков. Работа с закладками: скрипты, костюмы, звуки

1.2. Опыты с растровой графикой и векторной графикой.

Практика: Пиксель. Растр. Инструменты графического редактора. Особенности растровой графики. Создание фона для сцены и костюма для спрайта с помощью растрового графического редактора. Особенности векторной графики. Инструменты векторного редактора. Создание фона для сцены и костюма для спрайта с помощью растрового графического редактора.

1.2. Циклы. Опыты с блоками Движения.

Теория: Понятие цикла. Блоки Движение.

Практика: Непрерывное движение спрайта по сцене.

1.4. Начальная расстановка спрайтов. Опыты с блоками Внешность.

Теория: Появление спрайта всегда в одном месте при запуске программы. Блоки Внешность.

Практика: Работа с костюмами и эффектами спрайта.

1.5. События. Проект «Начинающий музыкант».

Теория: События, как воздействовать на спрайта через нажатие кнопок клавиатуры.

Практика: Создание проекта «Начинающий музыкант».

1.6. Опыты с числами. Координаты.

Теория: Координаты. Переменная.

Практика: Создание скриптов для перемещения спрайтов по координатам. Работа с числами.

1.7. Опыты со словами. Вращение и градусы.

Теория: «Операторы». Переменная. Списки. Вращения.

Практика: Создание скриптов для обработки слов и списков. Вращение спрайта.

Раздел 2. Эксперименты с блоками

2.1. Сообщения. Эксперимент «Фрукты в корзину», Эксперимент «Волшебное зелье».

Теория: Создание сообщений.

Практика: Сообщения. Отправка и получение сообщений. Метод последовательной детализации. Циклы. События. Последовательное и параллельное выполнение скриптов. Создание анимационного проекта.

2.2. Условия и операторы выбора. Эксперимент «Выключаем звезды»

Теория: Создание блоков. Работа со сценой. Условия и операторы. Отладка.

Практика: Условные операторы. Практические работы: «Выключаем звезды», «Диалоги», «Сцены».

2.3. Изменение координат. Процедуры. Эксперимент «Лабиринт».

Теория: Перемещение по заданным координатам.

Практика: Переход на другой уровень с помощью смены сцен. Разбиение программы на процедуры. Сборка программы из процедур. Практические работы: «Лабиринт», «Пинг-понг».

2.4. Логические операторы. Цикл с условием. Эксперимент «Кнопка лифта»

Теория: Операторы И, ИЛИ, НЕ. Переменные. Условия выполнения цикла.

Практика: Практическая работа «Кнопка лифта».

2.5. Случайные числа и диапазоны. Эксперимент «Дождь».

Теория: Диапазон чисел, выдать случайное.

Практика: Игральный кубик в Скретч, практическая работа «Дождь».

2.6. Эксперимент «Чат-бот».

Практика: Создание «Чат-бота». Ввод данных с клавиатуры

2.7. Переменные в играх. Эксперимент «Счет».

Практика: Практическая работа «Счет».

Раздел 3. Основные понятия электроники

3.1. Знакомство с контроллером Ардуино.

Теория: Закон Ома для участка цепи. Законы параллельного и последовательного соединения проводников. Светодиоды. Резисторы. Основные принципы маркировки резисторов. Макетная доска. Чтение электрических схем. Управление светодиодом на макетной доске.

Микроконтроллеры в нашей жизни (сообщения учеников), контроллер, контролер Ардуино (сообщение учеников), структура и состав Ардуино. Среда программирования для Ардуино (IDE Arduino) и язык программирования. (Презентация 2.)

Практика: Определение номинала резистора по цветовому коду, с помощью мультиметра

3.2. Основы проектирования и моделирования электронного устройства на базе Ардуино.

Теория: Управление электричеством. Законы электричества. Как быстро строить схемы: макетная доска (breadboard). Чтение электрических схем. Управление светодиодом на макетной доске.

Практика: Управление светодиодом на макетной доске, светофор, мигающие светодиоды. Маячок, железнодорожный семафор, светофор (3 секции), азбука Морзе.

3.3. Широтно-импульсная модуляция.

Теория: Аналоговые и цифровые сигналы, понятие ШИМ, управление устройствами с помощью портов, поддерживающих ШИМ. Циклические конструкции, датчик случайных

чисел, использование датчика в программировании для Ардуино.

Практика: Мигание двумя светодиодами.

Маячок с нарастающей / убывающей яркостью, моделируем пламя свечи, стробоскоп, управляемый светильник.

3.4. Программирование Ардуино. Пользовательские функции.

Теория: Подпрограммы: назначение, описание и вызов, параметры, локальные и глобальные переменные.

Практика: «Все цвета радуги». Управление RGB-светодиодом, управление звуком, светом и цветом.

3.5. Сенсоры. Датчики Ардуино.

Теория: Роль сенсоров в управляемых системах. Сенсоры и переменные резисторы.

Делитель напряжения. Потенциометр. Аналоговые сигналы на входе Ардуино.

Использование монитора последовательного порта для наблюдений за параметрами системы.

Практика: Автоматическое освещение, Измерение температуры в помещении и на улице, измерение влажности, датчики освещенности.

3.6. Кнопка – датчик нажатия.

Теория: Особенности подключения кнопки. Устранение шумов с помощью стягивающих и подтягивающих резисторов. Программное устранение дребезга. Булевские переменные и константы, логические операции.

Практика: Светофор с секцией для пешеходов и кнопкой управления.

3.7. Цифровые индикаторы. Семисегментный индикатор.

Теория: Назначение, устройство, принципы действия семисегментного индикатора.

Управление семисегментным индикатором. Программирование: массивы данных.

Практика: Счёт до 10, обратный счёт, Секундомер.

3.8. Жидкокристаллический экран.

Теория: Назначение и устройство жидкокристаллических экранов. Вывод сообщений на экран.

Практика: Вывод сообщений и показаний на экран дисплея.

Раздел 4. Проектная деятельность

4.1. Кнопочный переключатель.

Практика: Проекты по готовым разработкам: создаем из тактовой кнопки триггер, борясь с «дребезгом».

4.2. Светильник с кнопочным управлением.

Практика: Проекты по готовым разработкам: создаем порцию яркости светодиоду одной кнопкой и убавляем другой.

4.3. Кнопочные ковбои.

Практика: Проекты по готовым разработкам: создаем игрушку на реакцию: кто быстрее нажмет кнопку по сигналу.

4.4. Счётчик нажатий.

Практика: Проекты по готовым разработкам: выводим на семисегментный индикатор количество нажатий на кнопку.

4.5. Комнатный термометр.

Практика: Проекты по готовым разработкам: измеряем температуру окружающей устройство среды и с помощью шкалы показываем, на сколько она превышает заданный порог.

4.6. Перетягивание каната.

Практика: Проекты по готовым разработкам: создаем еще одну игру, на этот раз нужно быстрее соперника нажать кнопку 20 раз.

4.7. Пантограф.

Практика: Проекты по готовым разработкам: вращаем сервопривод на угол, задаваемый потенциометром.

4.8. Создание проекта.

Практика: Создание проекта.

Результаты образовательного процесса обучения

<i>Будут знать</i>	<i>Будут уметь</i>
- технику безопасности; -основные конструкции языков программирования; -приемы работы в средах программирования, подробно в среде Scratch, текстовых, графических, звуковых редакторах, браузерах; -построение информационных моделей из различных предметных областей с использованием типовых средств; -отдельные способы планирования деятельности; - принципы программирования Arduino; -способы и приемы соединения деталей (комбинированные соединения, рациональную последовательность операций по сборке деталей); понимать заданные схемы («принципиальная схема» и «схема на макете») электронных устройств и воспроизводить их на макетной плате; – понимать назначение элементов, их функцию; - понимать правила соединения деталей в единую электрическую цепь.	– выбирать и запускать программную среду Scratch; – работать с основными элементами пользовательского интерфейса программной среды; – создавать игры; – создавать мультфильмы; – использовать меню «быстрых» клавиш, кнопок в окнах диалога, шрифтов; – сформулировать тематику проекта и выполнить проект использовать готовые информационные модели, оценивать их соответствие реальному объекту и целям моделирования; - осуществлять выбор способа представления информации в соответствии с поставленной задачей; - иллюстрировать учебные работы с использованием средств информационных технологий; - создавать информационные объекты сложной структуры ; - осуществлять поиск информации в компьютерных сетях и пр.; - владеть основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности; - собрать и подключить к микроконтроллеру проект по предложенной схеме; -подключать датчики, двигатели и прочие элементы, необходимые для сборки проекта; -осуществлять программирование моделей по заданной схеме, по поставленной задаче; -осуществлять взаимный контроль и оказывать необходимую взаимопомощь; - создавать и преобразовывать модели и схемы для решения задач; - владеть основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности; -организовывать учебное сотрудничество и совместную деятельность с педагогом и сверстниками.

Карта самооценки и оценки педагогом компетентности учащегося

Учащемуся предлагается оценить по пятибалльной шкале полученные в процессе обучения знания и умения. Выдается бланк со следующими вопросами:

1. Освоил теоретический материал по разделам и темам программы (могу ответить на вопросы педагога).
2. Знаю специальную терминологию, используемую на занятиях.
3. Научился использовать полученные на занятиях знания в практической деятельности.
4. Умею выполнять практические задания (практические работы, упражнения, задачи), которые дает педагог.
5. Научился самостоятельно выполнять задания.

6. Умею технически реализовывать задуманные идеи.
7. Могу научить других, тому чему научился сам на занятиях.
8. Научился сотрудничать и взаимодействовать с другими ребятами в решении поставленных задач.
9. Научился получать информацию из других источников.
10. Мои достижения и результаты занятий.

Методическое обеспечение программы

При реализации программы важно соблюдать оптимальное сочетание традиционных методов обучения с новыми информационными технологиями. Предполагается использовать следующие технологии обучения:

1. Информационные, компьютерные (вытекающие из специфики программы).
2. Игровые технологии (организация учебного процесса осуществляется с учётом возрастных особенностей, ведущего типа деятельности).
3. Проблемно-поисковые технологии (ведущий метод – метод проектов). Организация учебного процесса осуществляется с учётом структуры любого вида сознательной деятельности человека: мотив (цель, замысел), ориентировка, планирование, реализация замысла, контроль.
4. Здоровьесберегающие технологии (соблюдение установленных норм и правил организации рабочего места, соблюдение регламентированного времени работы учащихся за компьютером, педагогически целесообразный баланс между традиционными методами преподавания и включением в учебно-воспитательный процесс информационных технологий).

Курс носит сугубо практический характер, поэтому центральное место в программе занимают практические умения и навыки работы на компьютере. Понятия и термины вводятся постольку, поскольку они необходимы для формирования названных умений и навыков.

Теоретические понятия информатики изучаются в традиционной форме – объяснение педагога с вовлечением учащихся в диалог. Используются формы исследовательского, проблемного обучения с групповым обсуждением учебных и практических задач. Изучению теоретических понятий отводится 15-20 минут занятия.

Оставшееся время используется для практических заданий, которые выполняются учащимися самостоятельно под контролем педагога. Из этого времени работе на ПК отводится не более 15 минут. В настоящее время количество компьютеров позволяет каждому ребенку работать за отдельной машиной, поэтому предполагается работа в индивидуальном режиме.

Программный материал изучается на базовом уровне с дифференциацией практических заданий для детей, быстрее освоивших предыдущие темы.

Основной тип занятия – комбинированный.

Беседы с детьми сопровождаются показом иллюстраций, схем, готовых шаблонов, изделий с использованием современных визуальных технических средств обучения.

Проверка знаний по программе учащихся построена на выполнении заданий, отражающих уровень понимания ими теоретических знаний, а также творческих и практических заданий (Приложение 1).

Формы и критерии оценки результативности определяются самим педагогом и заносятся в протокол (бланк ниже), чтобы можно было определить отнесенность обучающихся к одному из трех уровней результативности: высокий, средний, низкий (Приложение 2, 3, 4, 5, 6).

Программное обеспечение курса

Материально-техническая база позволяет реализовывать курс в компьютерном классе, и всегда выполняются следующие условия:

1. Каждый учащийся на каждом уроке обеспечен компьютерным рабочим местом, специально оборудованным в соответствии с его возрастом.
2. Учитель имеет свое компьютерное рабочее место.
3. На жестких дисках выделено достаточное дисковое пространство для хранения работ учащихся.
4. Каждое компьютерное рабочее место оборудовано компьютером под управлением ОС Windows 10.
5. В набор программного обеспечения каждого компьютера входит стандартный набор программ MS Office, приложение Scratch.
6. Все компьютеры класса включены в локальную сеть и имеют (локальный) доступ к серверу.
7. В учебном классе находится принтер и сканер, присоединенные к локальной сети.
8. Техническими средствами и инструментами (конструкторами) на каждого учащегося:

№	Наименование	Кол-во
1	Arduino UNO с кабелем	1
2	Breadboard на 830 точек	1
3	Кнопочная матрица 4x4	1
4	Символьный дисплей 1602 с I2C переходником	1
5	Провода для прототипирования	1
6	Модуль часов реального времени DS1302	1
7	Датчик температуры и влажности DHT-11	1
8	Модуль RFID RC522 с картой и брелком	1
9	Модуль реле 5В	1
10	Датчик уровня влаги	1
11	Датчик шума аналоговый + цифровой	4
12	Модуль джойстика KY-023	1
13	Модуль RGB светодиода	1
14	ИК пульт	1
15	ИК приемник TL1838	1
16	Шаговый двигатель 28YBJ-48	1
17	Драйвер на ULN2003	1
18	Сервопривод Tower Pro SG90	1
19	Датчик вибрации	1
20	Датчик температуры LM35DZ	1

СПИСОК ЛИТЕРАТУРЫ

1. Изучаем Arduino/ Блум Джереми. // Санкт-Петербург: изд. БХВ-Петербург, 2015. – С.334
2. Постановление администрации ЗАТО г. Североморск № 556 от 01.04.2020 г. «Об утверждении Положения о персонифицированном дополнительном образовании в ЗАТО г. Североморск».
3. Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 г. № 28 «Об утверждении санитарных правил СП 2.4. 3648-20 "Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления учащихся и молодежи».
4. Постановление Главного государственного санитарного врача Российской Федерации от 28 января 2021 года № 2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».
5. Приказ Министерства просвещения РФ от 27 июля 2022г. N629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
6. Програмируем Arduino. Основы работы со скетчами/ Монк С.//Санкт- Петербург: изд. Питер, 2016. – С.176.
7. Проекты с использованием контроллера Arduino/ Петин В.// - Санкт-Петербург: изд. БХВ-Петербург, 2015. – С.448.
8. Распоряжение Правительства Мурманской области от 03.07.2019 № 157-РП «О концепции внедрения целевой модели региональной системы дополнительного образования учащихся в Мурманской области».
9. Устав Муниципального бюджетного учреждения дополнительного образования ЗАТО г. Североморск «Дом детского творчества им. Саши Ковалева».
10. Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
11. Информатика. Курс лекций. Учебник / О.С. Логунова // Лань. - 2018. - С.148.
Информатика. Практические задания. Учебное пособие / И.В. Орлова // Лань. – 2019. – С.140.
12. Компьютерный дизайн. Векторная графика. Учебно-методическое пособие / Е.А. Зиновьева // Флинта. - 2020. – С.116.
13. Компьютерная графика в дизайне / Ф.Д.Миронов // ВНУ. – 2014. – С.560.
14. Электроника для начинающих / Платт Чарльз // - Санкт-Петербург: изд. БХВ-Петербург, 2012. – С.480.

Литература для педагога

1. Учимся создавать игры на Scratch программирование для детей/Ю.Торгашева // Питер. -2018. - С.128.
2. Мои первые программы на Scratch /Ю.Торгашева // Питер. -2018. - С.95.
3. Рисуем на компьютере / А.А Дуванов // БХВ-Петербург - 2005. – С.170.
4. Програмируем на Scratch. Приключения в космосе/Уэйнрвйт, Макс // Клевер-Медиа-Групп. -2018. – С31.
5. 40 проектов на Scratch для юных программистов/Д.Голиков// БХВ-Петербург. -2019. - С.192.
6. Scratch и Arduno для юных программистов и конструкторов /Ю.Виницкий, А.Григорьев// БХВ-Петербург. -2019. - С.176.
7. Основы программирования микроконтроллеров/ Артём Бачинин, Василий Панкратов, Виктор Накоряков// М.: Амперка, 2013г. - 205 с.

8. Програмуємо на Scratch. Приключения в джунглях/Уэйнрйт, Макс // Клевер-Медиа-Групп. -2018. – С31.

Литература для учащихся

1. Scratch для юных программистов /Д.Голиков // БХВ-Петербург. -2019. - С.192.
2. Практическая энциклопедия Arduino/ Петин, Биняковский// [ДМК-Пресс](#), 2020 г.- С.166.
3. Программирование микроконтроллерных плат Arduino/Freduino/ У.Соммер// [ВНУ](#), 2016 г.- С.256.
4. Путешествие в страну компьютерной графики / С.Скрылина // БХВ-Петербург. – 2014. - С.122.
5. Электроника. Занимательная электроника/ Ю/ Ревич// ВНУ, 2021 г. – С.688.

Приложение 1

Виды контроля

Виды контроля	Содержание	Методы	Сроки контроля
Предварительный	Начальный уровень подготовки учащихся, имеющиеся знания, умения и навыки, связанные с предстоящей деятельностью.	Опрос	Сентябрь
Текущий	Освоение учебного материала по темам.	Кроссворды, ребусы Задания, тесты	В течение года
Промежуточный	Освоение учебного материала за полугодие	Опросы, практические задания	Декабрь-январь апрель
Итоговый	Освоение учебного материала за год	Защита творческого задания	апрель

Приложение 2

Оценка уровней освоения программы

Уровни / количество %	Параметры	Критерии	Показатели
Высокий уровень/ 100-80%	Теоретические знания.	Широта кругозора, свобода восприятия теоретической информации, развитость практических навыков работы со специальной литературой, осмысленность и свобода использования специальной терминологии.	Учащийся освоил практически весь объём знаний 100-80%, предусмотренных программой за конкретный период; специальные термины употребляет осознанно и в полном соответствии с их содержанием
	Практические умения и навыки.	Соответствие развития уровня практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением, качество выполнения практического задания, технологичность практической деятельности.	Учащийся овладел на 100-80% умениями и навыками, предусмотренными программой за конкретный период; работает с оборудованием самостоятельно, не испытывает особых трудностей; выполняет практические задания с элементами творчества
Средний уровень/ 79-50%	Теоретические знания.	Широта кругозора, свобода восприятия теоретической информации, развитость практических навыков работы со специальной литературой, осмысленность и свобода использования специальной терминологии.	У учащегося объём усвоенных знаний составляет 79-50%; сочетает специальную терминологию с бытовой
	Практические умения и навыки.	Соответствие развития уровня практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением, качество выполнения практического задания, технологичность	У учащегося объём усвоенных умений и навыков составляет 79-50%; работает с оборудованием с помощью педагога; в основном, выполняет задания на основе образца

		практической деятельности.	
Низкий уровень / Менее 50%	Теоретические знания.	Широта кругозора, свобода восприятия теоретической информации, развитость практических навыков работы со специальной литературой, осмысленность и свобода использования специальной терминологии.	Учащийся овладел менее чем 50% объёма знаний, предусмотренных программой; учащийся, как правило, избегает употреблять специальные термины.
	Практические умения и навыки.	Соответствие развития уровня практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением, качество выполнения практического задания, технологичность практической деятельности.	Учащийся овладел менее чем 50%, предусмотренных умений и навыков; испытывает серьёзные затруднения при работе с оборудованием; обучающийся в состоянии выполнять лишь простейшие практические задания педагога.

Промежуточная диагностика

Программа _____
 Педагог д/о _____
 Группа № _____ год обучения _____
 Уровень теоретических знаний и / или _____
 Уровень практических умений и навыков _____
 Форма проведения _____

№	Ф.И. учащегося	%
1.		
2.		
12.		

Средний %

Задание

Разработать озвученную анимированную историю, выполненную в среде программирования Scratch, по заданной организаторами теме. Проект представляет собой анимационный ролик с озвучкой или игра.

Критерии оценивания творческого задания учащегося

№	Содержание задания	Количество мероприятий/ баллы	Максимальное количество баллов	Срок предоставления
1	Собранная схема на макетной плате Arduino UNO	- проект работает автономно, с небольшим вмешательством человека (0-20); - проект работает автономно, с вмешательством человека (0-10); - не использованы дополнительные команды (0);	30 баллов	Апрель
	Программирование	- проект в среде программирования выполнен аккуратно, с нестандартным оригинальным подходом авторов (0-10); - проект не имеет оригинальности и выполнен небрежно (0). - программа написана грамотно, выполнение происходит логично на основе ввода данных с датчиков (0-30);	40 баллов	Апрель
	Защита проекта	- прослеживается логика и последовательность изложения рассказа (0-10); - перечислены инструменты, использованные при создании игры (0-10); - грамотно отвечает на поставленные вопросы, используя терминологию (0-10)	30 баллов	Апрель
		Итого:	100	

Обучающийся, набравший более 50 баллов, считается успешно завершившим обучение.

Промежуточная аттестация обучающихся после изучения раздела «Эксперименты с блоками»

Форма проведения: тестирование, практическая работа.

Тестирование

Задание: выбрать один правильный вариант ответа из предложенных.

Критерий оценки: За каждый правильный ответ – 1 балл.

За неправильный ответ или отсутствие ответа – 0 баллов.

Максимальное количество баллов за тестирование – 5. 1.

1. Место, где происходят главные события в Scratch, называется:

- А) список спрайтов
- Б) палитра блоков
- В) сцена

2. Блоки команд в Scratch отличаются:

- А) размером
- Б) цветом
- В) способом использования

3. Изображения, действующие в Scratch программе, называются:

- А) скрипт
- Б) спрайт
- В) аниматор

4. Слово СКРИПТ на русском языке означает:

- А) сценарий
- Б) программа
- В) анимация

5. Диапазон координаты X в Scratch:

- А) от -240 до 240
- Б) от -180 до 180
- В) от -384 до 384

Ключ к тесту: 1 В

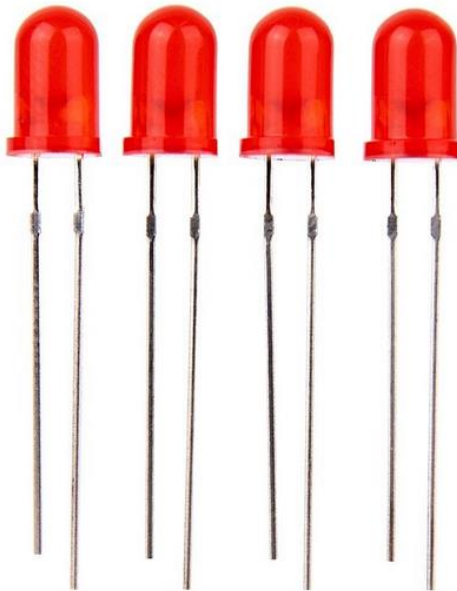
1	В
2	Б
3	Б
4	А
5	А

Промежуточная аттестация обучающихся после изучения раздела «Основные понятия электроники»

Форма проведения: тестирование, практическая работа.

Тестирование

Какова правильная полярность подключения светодиода? *



- ☐ Длинная ножка (анод) к «минусу» питания, короткая ножка (катод) – к «плюсу»
- ☐ Длинная ножка (катод) к «плюсу» питания, короткая ножка (анод) – к «минусу»
- ☒ Длинная ножка (анод) к «плюсу» питания, короткая ножка (катод) – к «минусу»

В электронике **анод** — электрод, через который ток втекает в элемент, **катод** — электрод, через который ток вытекает.

Вообще, у всех «потребителей» электричества анодом называют контакт, подключаемый к плюсу.

В качестве мнемонического правила, чтобы это запомнить, можно использовать такое:

— «анод» — 4 буквы, «плюс» — 4 буквы => анод подключается к плюсу;

— «катод» — 5 букв, «минус» — 5 букв => катод подключается к минусу.

В чем необходимо обязательно убедиться перед загрузкой программы в контроллер Arduino? *

- ☒ Выбран тип платы
- ☐ В коде созданы макроопределения
- ☒ Плата физически подключена к компьютеру
- ☒ Выбран порт, к которому подключена плата

Для назначения режима работы пинов Arduino используется: *

- ☐ директива #define
- ☒ функция pinMode()
- ☐ функция digitalWrite()
- ☐ функция digitalRead()

pinMode() — устанавливает режим работы заданного вход/выхода(pin) как входа или как выхода.

Синтаксис:

pinMode(pin, mode)

Параметры:

pin: номер вход/выхода(pin), который Вы хотите установить

mode: режим одно из двух значение — INPUT или OUTPUT, устанавливает на вход или выход соответственно.

Процедура void setup() выполняется *

- ☐ только один раз
- ☒ один раз при включении платы Arduino
- ☐ все время, пока включена плата Arduino

Функция setup() вызывается, когда стартует скетч. Setup() запускает только один раз, после каждой подачи питания или сброса платы Arduino.

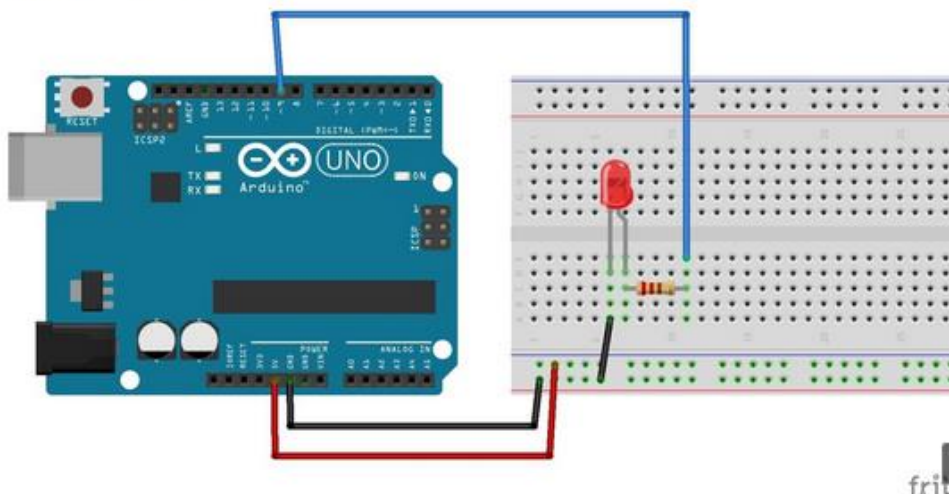
Используется для инициализации переменных, определения режимов работы выводов, запуска используемых библиотек и т.д.

Как работает «=»? *

- ☐ Это оператор сравнения
- ☒ Это оператор присваивания, он помещает значение, расположенное справа от него, в переменную, стоящую слева
- ☐ Это оператор присваивания, он делает оба операнда равными большему из них

В общем случае, оператор присваивания «=» присваивает переменной слева от оператора значение переменной или выражения, находящееся справа

Для какой цели в данной схеме используется резистор, последовательно соединенный со светодиодом? *



- ☒ Для уменьшения силы тока, текущего через светодиод
- ☐ Для увеличения яркости свечения светодиодов
- ☐ Для увеличения силы тока, текущего через светодиод
- ☐ Для подавления шума на выводе кнопки

После насыщения, при подаче напряжения выше некоторого порогового значения, собственное сопротивление светодиода становится очень мало, и без резистора, ограничивающего ток через светодиод, он перегорит.

Функция `delay()` *

- ☒ останавливает выполнение программы на заданное количество миллисекунд
- ☐ останавливает мигание светодиода на заданное количество миллисекунд
- ☐ останавливает выполнение программы на заданное количество секунд

`delay()` — останавливает выполнение программы на заданное в параметре количество миллисекунд.

Синтаксис:

`delay(ms)`

Параметры:

ms: количество миллисекунд, на которое приостанавливается выполнение программы.

Для считывания значений с аналогового входа используется команда *

- ☐ `digitalRead();`
- ☐ `digitalWrite();`
- ☒ `analogRead();`
- ☐ `analogWrite();`
- ☐ `pinMode();`
- ☐ `delay();`

`analogRead()` — считывает значение с указанного аналогового входа.

Синтаксис:

`analogRead(pin)`

Параметры:

pin: номер порта аналогового входа с которого будет производиться считывание.

Для считывания значений с цифрового входа используется команда *

- ☒ `digitalRead();`
- ☐ `digitalWrite();`
- ☐ `analogRead();`
- ☐ `analogWrite();`
- ☐ `pinMode();`
- ☐ `delay();`

`digitalRead()` — считывает значение с заданного входа — HIGH или LOW..

Синтаксис:

`digitalRead(pin)`

Параметры:

pin: номер вход/выхода(pin), с которого будет производиться считывание.

В какой строчке нет ошибки? *

- ☒ if (push==1) digitalWrite(13,HIGH);
- ☐ if (push>1); digitalWrite(13,HIGH);
- ☐ if (push>=1) digitalRead(13,1);
- ☐ if (push>=1) analogRead(13,500);

В третьем и четвертом варианте ошибки в вызове функций **digitalRead()** и **analogRead()** соответственно.

Во втором варианте код может скомпилироваться, но ошибка логическая. «;» после **if()** завершит обработку условного оператора, следующая команда выполнится независимо от того, выполнено условие или нет.

Что верно в отношении функции pinMode()? *

- ☐ В эту функцию можно не передавать параметры
- ☒ Принимает параметром направление работы порта (вход или выход)
- ☒ Принимает параметром номер пина, который конфигурируется
- ☒ Эта функция нужна для конфигурации направления работы порта

Что следует помнить при создании переменной? *

- ☒ Ей нужно задать тип
- ☒ Ей нужно выбрать имя
- ☒ Ей можно присвоить значение
- ☐ Имя состоит из латинских букв (обязательно начинается с нее), цифр и символов «_»
- ☒ Имя переменной нужно давать уникальное и осмысленное
- ☒ Это инструкция, должна заканчиваться «;»
- ☐ Значение переменной нельзя будет изменить

Переменная — имя для области в памяти, в которой хранятся данные. Переменная имеет имя, тип и значение.

В имени переменной допустимо использовать цифры, буквы, знак ‘_’. Нельзя начинать имя переменной с цифры.

Желательно, чтобы имя переменной несло смысловую нагрузку.

Что верно в отношении функции digitalWrite()? *

- ☐ В эту функцию можно не передавать параметры
 - ☒ Принимает параметром номер пина, которым нужно управлять
 - ☒ Эта функция позволяет включать или выключать напряжение на определенном пине
 - ☐ В качестве выставляемого напряжения можно указать любое напряжение в диапазоне 0—5В
 - ☒ Принимает параметром уровень напряжения (высокий или низкий), который необходимо выставить на контакте
 - ☒ Уровень напряжения можно задать константами HIGH (напряжение питания, 5В для Arduino UNO) и LOW (0В)
-

Что означает появившаяся после компиляции программы ошибка «'PIN_1' was not declared in this scope» *

- ☐ не закрыта скобка или нет точки запятой после PIN_1
- ☒ в скетче не объявлена переменная PIN_1
- ☐ в функции pinMode() не использовано имя порта PIN_1

Дословно: *PIN_1 не был объявлен в этой области.*

Какие утверждения относятся к условному оператору if? *

- ☒ условием может быть логическое выражение
- ☐ внутри else нельзя использовать другой if
- ☐ внутри if нельзя использовать другой if
- ☒ с помощью него можно задать условие, в зависимости от которого определенные действия будут или не будут выполнены
- ☒ else позволяет определить действия, которые выполняются, если условие ложно

Синтаксис:

```
if (<условие>)  
    { <действия1>; }  
else  
    { <действия2>; }
```

Условный оператор выполняет определенный набор команд, если некоторое логическое выражение (условие) истинно, и другой набор команд, если оно ложно.

Каким образом измеряется сила тока с помощью мультиметра? *



- ☐ Нужно включить мультиметр в режим прозвонки
- ☒ Нужно выбрать диапазон измерений (предполагаемую верхнюю границу)
- ☒ Нужно включиться щупами в цепь последовательно
- ☐ Нужно извлечь из мультиметра батарею
- ☒ Нужно установить щуп в разъем мультиметра, соответствующий предполагаемым токам

К чему приведет выполнение следующего кода? *

```
1 void setup() {  
2   pinMode(2, OUTPUT);  
3   pinMode(3, OUTPUT);  
4  
5   digitalWrite(2, LOW);  
6   digitalWrite(3, LOW);  
7  
8 }  
9  
10 void loop() {  
11   digitalWrite(2, HIGH);  
12   digitalWrite(3, HIGH);  
13 }
```

- ☐ Напряжение на 2 и 3 пине будет включаться и выключаться
- ☐ Будет включено напряжение на 2 пине, затем оно будет выключено и включено на 3
- ☒ Будет включено напряжение на 2 пине, затем на 3 пине
- ☐ Будет включено напряжение на 2 пине, затем оно будет выключено и включено на 3, на следующей итерации loop() напряжение выключится на 3 пине и вновь включится на 2

При старте скетча, сначала вызывается функция setup(), в которой устанавливаются режимы работы 2 и 3 пинов, после чего на них устанавливается напряжение 0В. После этого запускается цикл loop(), в котором на пинах 2 и 3 последовательно включается напряжение 5В.