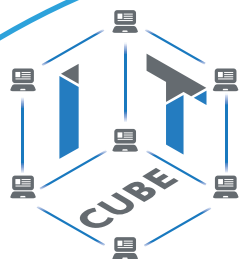




МИНИСТЕРСТВО
ПРОСВЕЩЕНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ

ОБРАЗОВАНИЕ

НАЦИОНАЛЬНЫЕ
ПРОЕКТЫ
РОССИИ



СЕТЬ ЦЕНТРОВ ЦИФРОВОГО
ОБРАЗОВАНИЯ ДЕТЕЙ «IT-КУБ»

РЕАЛИЗАЦИЯ
ДОПОЛНИТЕЛЬНОЙ
ОБЩЕОБРАЗОВАТЕЛЬНОЙ
ПРОГРАММЫ ПО ТЕМАТИЧЕСКОМУ
НАПРАВЛЕНИЮ

«ПРОГРАММИРОВАНИЕ РОБОТОВ»

С ИСПОЛЬЗОВАНИЕМ
ОБОРУДОВАНИЯ ЦЕНТРА
ЦИФРОВОГО ОБРАЗОВАНИЯ
ДЕТЕЙ «IT-КУБ»

МОСКВА 2021



М.В. Курносенко

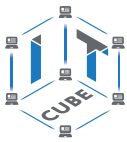
И.И. Мацаль

**Реализация дополнительной общеобразовательной программы по
тематическому направлению «Программирование роботов» с
использованием оборудования центра цифрового образования детей «IT-
куб».**

Методическое пособие

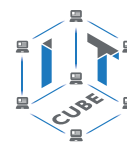
под ред. С. Г. Григорьева

Москва, 2021



Содержание

Пояснительная записка	3
Цель и задачи	4
Нормативная база	5
Основные понятия и термины	6
Структурирование материалов	7
Описание материально-технической базы центра цифрового образования детей «ИТ-куб»	8
Каталог оборудования	9
Примерная рабочая программа для организации работы	12
Тематическое планирование	14
Содержание и форма организации учебных занятий	16
Планы учебных занятий	17
Урок 1	17
Уроки 2–3	22
Уроки 4–5	27
Уроки 6–7	31
Уроки 8–9	35
Уроки 10–11	39
Уроки 12–15	42
Уроки 16–17	46
Урок 18	49
Урок 19	53
Урок 20	56
Урок 21	59
Урок 22	63
Урок 23	66
Урок 24	69
Урок 25	72
Урок 26	75
Урок 27	78
Урок 28	81
Урок 29	84
Урок 30	87
Урок 31	90
Урок 32	93
Урок 33	97
Урок 34	99
Урок 35	102
Урок 36	105
Приложение	108
Регламенты соревнований для роботов	108



Пояснительная записка

Сегодня потребность в программировании роботов стала такой же повседневной задачей для продвинутого учащегося, как решение задач по математике или выполнение упражнений по русскому языку. Существующие среды программирования, как локальные, так и виртуальные, служат хорошим инструментарием для того, чтобы научиться программировать роботов. Хотя правильнее сказать не роботов, а контроллеры, которые управляют роботами. Но «робот» — понятие более широкое, чем мы привыкли считать.

Справочник!

Робот — это любое электронное устройство, управляемое контроллером, который нужно соответствующим образом запрограммировать.

Для того чтобы запрограммировать робота, сначала необходимо сформировать у учащегося основы алгоритмического мышления. Для решения этой задачи лучше всего подходит популярная среда Scratch с графическим интерфейсом (<http://scratch.mit.edu>), которая наглядна и проста и, что немаловажно, бесплатна. В этой среде можно работать как в режиме онлайн (прямо на сайте), так и локально, установив редактор Scratch на свой ПК. Это позволит научить обучающихся программировать (создавать) игровые программы и тем самым получить ключевые навыки программирования на этом языке, которые в дальнейшем понадобятся для программирования роботов.

На следующем этапе, в зависимости от учебных планов и оборудования, можно начинать программировать уже конкретные устройства, как виртуальные, так и реальные, в частности роботов или электронные устройства (например, «умный дом»).

Самый простой способ запрограммировать робота в Scratch описан на сайте <https://vr.vex.com> («Виртуальные роботы VEX»), который также бесплатен. Здесь пользователь познакомится с датчиками и расширенными опциями движения. Представленный на этом интернет-ресурсе набор заданий (игровых полей или карт) для робота уже достаточно широк и может активно использоваться в учебном процессе.

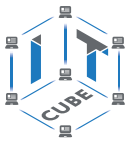
Программная среда Scratch является универсальной для программирования многих образовательных робототехнических систем (конструкторов), и поэтому выбор бесплатной платформы VEXcode VR обусловлен именно этими факторами.

Справочник!

Для совершенствования навыков работы со Scratch можно использовать следующие реальные образовательные робототехнические системы (конструкторы).

1. Цифровая лаборатория школьника «Тетра»: <https://amperka.ru/product/tetra-kit>.
2. Робоплатформа «Роббо»: <https://robbo.ru>.
3. Modkit for VEX: <http://vex.examen-technolab.ru/vexiq/iqprogrammirovanie>.
4. Lego Education Spike: <https://education.lego.com/ru-ru/products/-lego-education-spike-prime/45678#spike%E2%84%A2-prime>.

Подчеркнём, что многие производители робототехнических систем (VEX, «Роботрек» и пр.) так или иначе используют в своих редакторах кода программирование контроллеров с помощью графических блоков по аналогии со Scratch. Это упрощает переход уже



на «взрослое» программирование на других языках, чаще всего на языке Си. Во многих системах переход Scratch → Си происходит автоматически, т. е. программа, написанная в Scratch, автоматически переводится в Си, и наоборот.

После того как обучающиеся освоят программирование на Scratch, можно переходить к программированию на других языках, как было уже сказано выше, прежде всего, на язык Си, так как он является основным для программирования контроллеров, в первую очередь Arduino. В этом случае может помочь бесплатная среда онлайн-моделирования Tinkercad (<http://tinkercad.com>).

Цель и задачи

Цель программы «Программирование роботов»: развитие алгоритмического мышления обучающихся, их творческих способностей, аналитических и логических компетенций, а также пропедевтика будущего изучения программирования роботов на одном из современных языков.

Для формирования поставленной цели планируется достижение следующих **результатов**.

Личностные результаты:

- развитие пространственного воображения, логического и визуального мышления, наблюдательности, креативности;
- развитие мелкой моторики рук;
- формирование первоначальных представлений о профессиях, в которых информационные технологии играют ведущую роль;
- воспитание интереса к информационной и коммуникационной деятельности.

Метапредметные результаты:

- формирование алгоритмического мышления через составление алгоритмов в компьютерной среде VEXcode VR;
- овладение способами планирования и организации творческой деятельности.

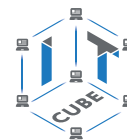
Предметные результаты:

- ознакомление с основами робототехники с помощью универсальной робототехнической платформы VEXcode VR или аналогичной ей (виртуальной или реальной);
- систематизация знаний по теме «Алгоритмы» на примере работы программной среды Scratch с использованием блок-схем программных блоков;
- овладение умениями и навыками при работе с платформой (конструктором), приобретение опыта практической деятельности по созданию автоматизированных систем управления, полезных для человека и общества;
- знакомство с законами реального мира;
- овладение умением применять теоретические знания на практике;
- усвоение знаний о роли автоматизированных систем управления в преобразовании окружающего мира.

При работе с платформой VEXcode VR решаются следующие основные **задачи**.

Познавательные задачи:

- начальное освоение компьютерной среды Scratch в качестве инструмента для программирования роботов;
- систематизация и обобщение знаний по теме «Алгоритмы» в ходе создания управляющих программ в среде Scratch;



- создание завершённых проектов с использованием освоенных навыков структурного программирования.

Регулятивные задачи:

- формирование навыков планирования — определения последовательности промежуточных целей с учётом конечного результата;
- освоение способов контроля в форме сопоставления способа действия и его результата с заданным образцом с целью обнаружения отличий от эталона.

Коммуникативные задачи:

- формирование умения работать над проектом в команде;
- овладением умением эффективно распределять обязанности.

Возраст обучающихся, участвующих в реализации программы: 7—14 лет.

Уровень освоения: программа является общеразвивающей (базовый уровень), не требует предварительных знаний и входного тестирования.

Режим занятий: занятия проводятся в группах до 12 человек, продолжительность одного занятия — 45 минут.

Сроки реализации: общая продолжительность программы — 36 часов.

Нормативная база

Конституция Российской Федерации (принята всенародным голосованием 12.12.1993 с изменениями, одобренными в ходе общероссийского голосования 01.07.2020). — URL: http://www.consultant.ru/document/cons_doc_LAW_28399/ (дата обращения: 10.03.2021).

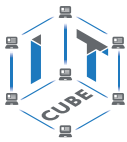
Федеральный закон от 29.12.2012 № 273-ФЗ (ред. от 31.07.2020) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 01.09.2020). — URL: http://www.consultant.ru/document/cons_doc_LAW_140174 (дата обращения: 28.09.2020).

Паспорт национального проекта «Образование» (утверждён президиумом Совета при Президенте РФ по стратегическому развитию и национальным проектам, протокол от 24.12.2018 № 16). — URL: http://www.consultant.ru/document/cons_doc_LAW_319308/ (дата обращения: 10.03.2021).

Государственная программа Российской Федерации «Развитие образования» (утверждена постановлением Правительства РФ от 26.12.2017 № 1642 (ред. от 22.02.2021) «Об утверждении государственной программы Российской Федерации «Развитие образования»). — URL: http://www.consultant.ru/document/cons_doc_LAW_286474/ (дата обращения: 10.03.2021).

Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена распоряжением Правительства РФ от 29.05.2015 № 996-р «Об утверждении Стратегии развития воспитания в Российской Федерации на период до 2025 года»). — URL: http://www.consultant.ru/document/cons_doc_LAW_180402/ (дата обращения: 10.03.2021).

Профессиональный стандарт «Педагог (педагогическая деятельность в дошкольном, начальном общем, основном общем, среднем общем образовании), (воспитатель, учитель)» (ред. от 16.06.2019) (приказ Министерства труда и социальной защиты РФ от 18 октября 2013 г. № 544н, с изменениями, внесёнными приказом Министерства труда и соцзащиты РФ от 25 декабря 2014 г. № 1115н и от 5 августа 2016 г. № 422н). — URL: http://www.consultant.ru/document/cons_doc_LAW_155553/ (дата обращения: 10.03.2021).



Профессиональный стандарт «Педагог дополнительного образования детей и взрослых» (приказ Министерства труда и социальной защиты РФ от 5 мая 2018 г. № 298н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых»). — URL: https://profstandart.rosmintrud.ru/obshchiy-informatsionnyy-blok/natsionalnyy-reestr-professionalnykh-standartov/reestr-professionalnykh-standartov/index.php?ELEMENT_ID=48583 (дата обращения: 10.03.2021).

Федеральный государственный образовательный стандарт основного общего образования (утверждён приказом Министерства образования и науки Российской Федерации от 17 декабря 2010 г. № 1897) (ред. 21.12.2020). — URL: <https://fgos.ru> (дата обращения: 10.03.2021).

Федеральный государственный образовательный стандарт среднего общего образования (утверждён приказом Министерства образования и науки Российской Федерации от 17 мая 2012 г. № 413) (ред. 11.12.2020). — URL: <https://fgos.ru> (дата обращения: 10.03.2021).

Методические рекомендации по созданию и функционированию детских технопарков «Кванториум» на базе общеобразовательных организаций (утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-4). — URL: http://www.consultant.ru/document/cons_doc_LAW_374695/ (дата обращения: 10.03.2021).

Методические рекомендации по созданию и функционированию центров цифрового образования «ИТ-куб» (утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-5). — URL: http://www.consultant.ru/document/cons_doc_LAW_374572/ (дата обращения: 10.03.2021).

Методические рекомендации по созданию и функционированию в общеобразовательных организациях, расположенных в сельской местности и малых городах, центров образования естественно-научной и технологической направленностей («Точка роста») (утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-6). — URL: http://www.consultant.ru/document/cons_doc_LAW_374694/ (дата обращения: 10.03.2021).

Основные понятия и термины

Алгоритм — это конечное точное предписание действий, которые необходимо выполнить для решения поставленной задачи.

Исполнитель алгоритма — это некоторый объект (техническое устройство, робот, автомат), способный выполнять определённый набор команд алгоритма.

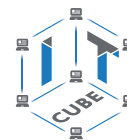
Робот — это исполнитель алгоритма, сформулированного на одном из языков программирования.

Среда Scratch — это среда программирования в виде графических блоков, описывающих команды исполнителю алгоритма.

Трансмиссия — это группа команд среды Scratch, задающих различные виды движений исполнителя алгоритма.

Датчик — это средство измерений, предназначенное для выработки сигнала измерительной информации в форме, удобной для передачи, дальнейшего преобразования, обработки и (или) хранения, но не поддающейся непосредственному восприятию наблюдателем.

Датчики, выполненные на основе электронной техники, называются **электронными датчиками**. Отдельно взятый датчик может быть предназначен для измерения (контро-



ля) и преобразования одной физической величины или одновременно нескольких физических величин.

Переменная (в императивном программировании) — это поименованная либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным. В таких языках переменная определяется как имя, с которым может быть связано значение, или даже как место (location) для хранения значения.

Игровое поле — это заранее сконфигурированная площадка с заданиями для робота.

Консоль экрана — это специальное окно для вывода значений и сообщений в ходе выполнения роботом заданий на игровом поле.

Структурирование материалов

Содержание обучения может быть представлено следующими модулями.

Модуль 1. Знакомство с платформой VEXcode VR.

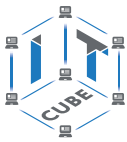
Модуль 2. Программирование робота на платформе.

Модуль 3. Датчики и обратная связь.

Модуль 4. Реализация алгоритмов движения робота.

Модуль 5. Творческий проект.

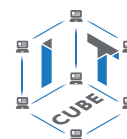
Модуль 6. Дальнейшее развитие.



Описание материально-технической базы центра цифрового образования детей «ИТ-куб»

Для организации работы «ИТ-куб» в распоряжении «Об утверждении методических рекомендаций по созданию и функционированию центров цифрового образования «ИТ-куб» от 12.02.2021 рекомендуется следующее оборудование лаборатории:

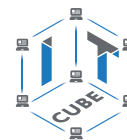
- ноутбук — рабочее место преподавателя;
- рабочее место обучающегося;
- жёсткая, неотключаемая клавиатура: наличие;
- русская раскладка клавиатуры: наличие;
- диагональ экрана: не менее 15,6 дюйма;
- разрешение экрана: не менее 1920 × 1080 пикселей;
- количество ядер процессора: не менее 4;
- количество потоков: не менее 8;
- базовая тактовая частота процессора: не менее 1 ГГц;
- максимальная тактовая частота процессора: не менее 2,5 ГГц;
- кеш-память процессора: не менее 6 Мбайт;
- объём установленной оперативной памяти: не менее 8 Гбайт;
- объём поддерживаемой оперативной памяти (для возможности расширения): не менее 24 Гбайт;
- объём накопителя SSD: не менее 240 Гбайт;
- время автономной работы от батареи: не менее 6 часов;
- вес ноутбука с установленным аккумулятором: не более 1,8 кг;
- внешний интерфейс USB стандарта не ниже 3.0: не менее трёх свободных;
- внешний интерфейс LAN (использование переходников не предусмотрено): наличие;
- наличие модулей и интерфейсов (использование переходников не предусмотрено): VGA, HDMI;
- беспроводная связь Wi-Fi: наличие с поддержкой стандарта IEEE 802.11n или современнее;
- веб-камера: наличие;
- манипулятор мышь: наличие;
- предустановленная операционная система с графическим пользовательским интерфейсом, обеспечивающая работу распространённых образовательных и общесистемных приложений: наличие;
- МФУ, веб-камера, интерактивный моноблочный дисплей, диагональ экрана: не менее 65 дюймов, разрешение экрана: не менее 3840 × 2160 пикселей, оборудованные напольной стойкой.



Каталог оборудования

Функционал оборудования	Внешний вид оборудования
<p>Виртуальная среда программирования роботов VEXcode VR.</p> <p>Предназначена для отработки навыков программирования роботов в среде Scratch и используется в дальнейшем при переходе на языки программирования Python и C++</p>	 <p>The image shows a VEX Code VR robot kit, which is a compact, orange and black robot with two large grey wheels and a smaller front wheel. It has a black sensor on top and a white marker. Below the robot is the VEX Code VR logo in yellow and black text.</p>
<p>Робототехнический конструктор с программируемым контроллером, комплектом датчиков и ресурсным набором комплектующих для разработки сложных мехатронных систем и моделей роботов для участия в робототехнических соревнованиях.</p> <p>Предназначен для разработки мобильных роботов и организации углублённой практики программирования. Программируется в редакторе RobotC как графическими блоками, так и в текстовом режиме. Может изучаться дистанционно в среде «Виртуальные миры»</p>	 <p>The image shows a VEX V5 robot kit, a more complex and larger robot with a grey and blue color scheme. It features a prominent blue gear mechanism on top, two large grey wheels, and a smaller front wheel. The robot is shown from a three-quarter perspective.</p>
<p>VEX V5 представляет собой пятое поколение образовательных робототехнических систем, разработанных с 20-летним опытом использования робототехники для обучения принципам STEM. Электроника V5 является доступной, гибкой и мощной, в ней используются самые современные технологии для обеспечения соответствующих результатов обучения.</p> <p>Механическая система V5 включает в себя универсальные элементы, которые делают проектирование доступным для начинающих пользователей, в то же время предоставляя опытным разработчикам безграничные возможности проектирования</p>	 <p>The image shows a VEX V5 robot kit, a complex robot with a grey and red color scheme. It has a large red gear mechanism on top, two large grey wheels, and a smaller front wheel. A black and grey game controller is shown in the foreground, connected to the robot.</p>

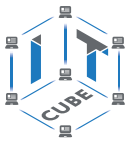
Функционал оборудования	Внешний вид оборудования
<p>Робот-манипулятор, разработанный и производимый в России, предназначен для освоения школьниками и студентами основ робототехники и подготовки к внедрению и последующему использованию роботов в промышленном производстве.</p> <p>В качестве управляющего контроллера применяется Arduino-совместимая плата, отлично зарекомендовавшая себя в линейке образовательных наборов для старшего школьного возраста. Благодаря такому подходу достигается методическая и программная совместимость с широко распространённым ПО mBlock. Оно обладает уникальными особенностями, позволяющими продуктивно работать с образовательным робототехническим оборудованием. Данное ПО основано на Scratch, но поддерживает и программирование на языке C, что существенно расширяет возрастные рамки для обучающихся, интересующихся программированием роботов</p>	
<p>Серия Lego Mindstorms EV3 разработана специально для обучения детей робототехнике в образовательных организациях, а также естественным наукам — физике, математике, информатике и технологии.</p> <p>Основу набора составляет микрокомпьютер Mindstorms EV3 с графическим дисплеем и портами для подключения датчиков. Сердцем набора является программируемый микрокомпьютер EV3, с помощью которого можно управлять роботом, контролировать работу моторов и датчиков, а также получать данные на компьютер посредством протоколов Bluetooth и Wi-Fi.</p> <p>Наборы Lego Mindstorms EV3 обладают высоким учебным потенциалом и могут быть использованы при обучении большинству предметов естественно-научного цикла для повышения эффективности учебного процесса</p>	



В инфографике, приведённой ниже, представлен временной ряд последовательного освоения программирования роботов от начального уровня с помощью виртуального робота до более сложных уровней — с помощью физических роботов VEX или им подобных.



Инфографика 1. Уровни освоения программирования роботов



Примерная рабочая программа для организации работы

Планируемые результаты освоения программы обучающимися

Личностные результаты:

- осмысление мотивов своих действий при выполнении заданий с жизненными ситуациями;
- начало профессионального самоопределения, ознакомление с миром профессий, связанных с информационными и коммуникационными технологиями.

Метапредметные результаты

I. Технологический компонент

Регулятивные УУД:

- освоение способов решения проблем творческого характера в жизненных ситуациях;
- формирование умений ставить цель — создание творческой работы, планирование достижения этой цели, создание вспомогательных эскизов в процессе работы;
- оценивание итогового творческого продукта и соотнесение его с изначальным замыслом, выполнение по необходимости коррекции либо продукта, либо замысла.

Познавательные УУД:

- поиск информации в индивидуальных информационных архивах учащегося, информационной среде образовательной организации, в федеральных хранилищах информационных образовательных ресурсов;
- использование средств информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач.

Коммуникативные УУД:

- подготовка выступления;
- овладение опытом межличностной коммуникации (работа в группах, выступление с сообщениями и т. д.).

II. Логико-алгоритмический компонент

Регулятивные УУД:

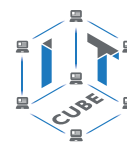
- планирование последовательности шагов алгоритма для достижения цели;
- поиск ошибок в плане действий и внесение в него изменений.

Познавательные УУД:

- моделирование — преобразование объекта из чувственной формы в модель, где выделены существенные характеристики;
- анализ объектов с целью выделения признаков (существенных, несущественных);
- синтез — составление целого из частей, в том числе самостоятельное достраивание с восполнением недостающих компонентов;
- установление причинно-следственных связей;
- построение логической цепи рассуждений.

Коммуникативные УУД:

- аргументирование своей точки зрения на выбор способов решения поставленной задачи;
- выслушивание собеседника и ведение диалога.



Предметные результаты

Модуль 1. Знакомство с платформой VEXcode VR

В результате изучения данного модуля учащиеся должны:

знать: названия различных компонентов робота и платформы: контроллер (специализированный микрокомпьютер); исполнительные устройства — мотор, колёса, перо, электромагнит; датчики цвета, расстояния, местоположения, касания; панель управления, курсы наблюдения робота; программные блоки по разделам; виды игровых полей (площадок); кнопки управления;

уметь: программировать управление роботом; использовать датчики для организации обратной связи и управления роботом; сохранять и загружать проект.

Модуль 2. Программирование робота на платформе

В результате изучения данного модуля учащиеся должны:

знать: математические и логические операторы; блоки вывода информации в окно вывода;

уметь: применять на практике логические и математические операции; использовать блоки для работы с окном вывода; составлять с помощью блоков математические выражения.

Модуль 3. Датчики и обратная связь

В результате изучения данного модуля учащиеся должны:

знать: принципы работы датчиков; блоки управления датчиками; возможности датчиков;

уметь: использовать циклы и ветвления для реализации системы принятия решений; решать задачу «Лабиринт».

Модуль 4. Реализация алгоритмов движения робота

В результате изучения данного модуля учащиеся должны:

знать: условный оператор if/else; цикл while; понятие шага цикла;

уметь: применять на практике циклы и ветвления; использовать циклы и ветвления для решения математических задач; использовать циклы для объезда повторяющихся траекторий.

Модуль 5. Творческий проект

При выполнении творческих проектных заданий учащиеся будут разрабатывать свои собственные программы. Проектные занятия могут проводиться учителем начальных классов, учителем технологии или учителем информатики.

Перечень используемого оборудования и материалов: рабочее место для работы с компьютером; компьютер с ОС Windows и выходом в Интернет; рабочая тетрадь ученика.

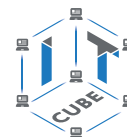
Модуль 6. Дальнейшее развитие

При выполнении задач учащиеся будут разрабатывать свои собственные программы. Проектные занятия могут проводиться учителем начальных классов, учителем технологии или учителем информатики.

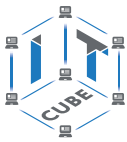
Перечень используемого оборудования и материалов: рабочее место для работы с компьютером; компьютер с ОС Windows и выходом в Интернет; рабочая тетрадь ученика.

Тематическое планирование

№ п/п	Тема	Содержание	Целевая установка урока	Кол-во часов	Основные виды деятельности обучающихся на уроке/внеурочном занятии	Использование оборудования
1	Модуль 1. Знакомство с платформой VEXcode VR	Основные фрагменты интерфейса платформы. Панель управления, блоки программы, датчики, игровая площадка, экран датчиков и переменных, кнопки управления. Создание простейших программ (скриптов), сохранение и загрузка проекта	Ознакомление обучающихся с интерфейсом платформы, принципами программирования виртуальной робота, видами игровых полей (площадок), основными блоками управления	3	Наблюдение за работой учителя, совместное с учителем программирование скриптов, самостоятельная работа с инструментами среды, ответы на контрольные вопросы	Виртуальная среда VEXcode VR
2	Модуль 2. Программирование робота на платформе	Математические и логические операторы, блоки вывода информации в окно вывода, блоки трансмиссии. Блоки управления, блоки переменных, блоки датчиков, блоки вида, магнит	Ознакомление обучающихся с блоками логических и математических операторов, приёмы работы с ними. Организация движения робота с помощью блоков трансмиссии. Применение блоков переменных. Изучение основных видов датчиков. Применение магнита	4	Наблюдение за работой учителя, совместное с учителем программирование скриптов, самостоятельная работа с инструментами среды, ответы на контрольные вопросы	Виртуальная среда VEXcode VR
3	Модуль 3. Датчики и обратная связь	Датчик местоположения, направления движения. Датчики цвета. Дискковый лабиринт. Датчик расстояния. Простой лабиринт. Динамический лабиринт.	Ознакомление обучающихся с основными видами датчиков и принципами их работы. Применение датчиков в различных игровых полях. Создание скриптов для прохождения простого и дина-	10	Наблюдение за работой учителя, совместное с учителем программирование скриптов, самостоятельная работа с инструментами среды, ответы на контрольные вопросы	Виртуальная среда VEXcode VR



4	Модуль 4. Реализация алгоритмов движения робота	Управление магнитом. Сбор фишек Блок команд «Управление» и организация циклов и ветвлений. Проекты «Разрушение замка» и «Динамическое разрушение замка». Проект «Детектор линии»	мического лабиринтов. Разработка программы сбора фишек с помощью магнита и размещение их по цветам	10	Наблюдение за работой учителя, совместное с учителем программирование скриптов, самостоятельная работа с инструментами среды, ответы на контрольные вопросы	Виртуальная среда VEXcode VR
5	Модуль 5. Творческий проект	Создание собственного проекта с использованием максимального возможного количества датчиков	На основе полученных знаний по работе с платформой каждый обучающийся создаёт свой проект	4	Наблюдение за работой учителя, совместное с учителем программирование скриптов, самостоятельная работа с инструментами среды, ответы на контрольные вопросы	Виртуальная среда VEXcode VR
6	Модуль 6. Дальнейшее развитие	Основы программирования роботов на языке Си. Простейшие программы для роботов	Используя полученные знания, обучающиеся знакомятся с принципами программирования роботов в текстовом редакторе RobotC на языке программирования Си	5	Наблюдение за работой учителя, совместное с учителем программирование скриптов, самостоятельная работа с инструментами среды, ответы на контрольные вопросы	Виртуальная среда VEXcode VR
	ИТОГО			36		



Содержание и форма организации учебных занятий

Рекомендуемый ход урока

Вступительная беседа

Вступительная беседа проводится с целью освежить в памяти учащихся накопленные знания и способы учебных действий, актуализировать их. Кроме того, необходимо психологически подготовить учащихся: сосредоточить их внимание на предстоящей деятельности, стимулировать интерес к уроку. В ходе беседы учащиеся воспроизводят известные им знания, осознают их, обобщают факты, связывают полученные ранее знания с новыми условиями, с новыми данными и т. д.

В процессе актуализации или в результате неё следует подвести учащихся к осознанию проблемной ситуации и формулированию проблемы. Этап актуализации должен подготовить учащегося к осуществлению самостоятельной учебной деятельности.

Формулирование темы

При формулировании темы следует обратить внимание учащихся на недостаточность формулировок типа «Датчик», поскольку при изучении модулей ставится задача уметь различать виды датчиков по принципу действия.

Тренировочное упражнение

Вначале учащимся предлагаются простые задачи, основной целью которых является выработка базовых навыков, таких как составление словесных описаний последовательностей действий робота, знакомство с функциями блоков управляющей программы, основами составления блок-схем, простыми управляющими программами.

На этом этапе предполагается групповое обсуждение задачи и способа её решения, возможна демонстрация фрагментов программы на интерактивной доске.

Тренировочные упражнения удобно выполнять до того, как решена основная задача из игрового поля.

После вступительной беседы и формулирования темы урока следует сразу приступить к работе с платформой. Необходимо научить учащихся вести сопроводительную документацию в рабочем блокноте, в будущем они будут работать с инженерной тетрадью. В рабочем блокноте они могут записывать алгоритмы, примеры программ, важные данные.

Учащиеся должны знать, что программе нужно присвоить имя и сохранить его на компьютере в папке для проектов. Работу за компьютером с платформой необходимо организовать индивидуально.

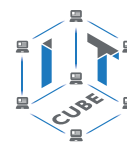
Самостоятельная работа

В ходе самостоятельной работы учащимся предлагается создать более сложную управляющую программу на базе полученных ранее знаний. Для того чтобы учащиеся успешно справились с этим видом деятельности, в инженерной тетради приводятся вспомогательные упражнения и подсказки, с которыми учащиеся работают самостоятельно.

Подведение итогов

В конце каждого урока полезно ещё раз проговаривать названия новых программных блоков, исполнительных устройств и датчиков, использованных при выполнении работы. Также следует выборочно проверять выполнение заданий в рабочей тетради учащихся.

Учитель предварительно знакомится с работами учащихся и выбирает 2–3 работы для демонстрации классу. Основная задача просмотра работ всем классом — отработать навык представления и защиты своего проекта, а также сформировать умение обсуждать и критически оценивать работу друг друга.



Планы учебных занятий

Урок 1

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 1. Знакомство с платформой VEXcode VR.

Тема урока: Основные фрагменты интерфейса платформы VEXcode VR. Панель управления, блоки программы, датчики, игровая площадка, экран датчиков и переменных, кнопки управления.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с платформой VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией платформы VEXcode VR (<http://vr.vex.com>), её инструментов и функциональных возможностей.

Платформа VEXcode VR облегчает процесс изучения информатики и робототехники, она позволяет учащимся кодировать виртуального робота, находясь в любом месте и программируя в среде блочного кодирования. Платформа VEXcode VR основана на VEXcode, том же интерфейсе программирования, который используется для робототехнических платформ VEX 123, GO, IQ и V5.

Показать удобство VEXcode VR для различных образовательных организаций.

Платформа VEXcode VR позволяет учащимся обогатить опыт использования компьютерных наук после того, как они открыли для себя программирование в среде Scratch и приобрели азарт работы с различными аппаратными платформами. VEXcode VR позволяет учащимся продолжать заниматься робототехникой, даже если поблизости нет физического робота.

Кроме того, VEXcode VR содержит функции, призванные помочь в освоении программирования и сделать STEM и информатику доступными для большего числа учащихся. Эти функции включают простоту использования, мгновенную обратную связь, возможность сделать обучение наглядным и помочь с различными видами творческой деятельности в классе.

Рассмотреть более подробно некоторые указанные функции и особенности платформы VEXcode VR.

Простота использования

VEXcode VR на 100 % основана на веб-технологиях, поэтому запустить платформу очень просто. Пользовательский интерфейс упрощает навигацию (команды разделены на категории), построен без использования большого списка команд в меню. Команды также имеют цветовую кодировку, поэтому пользователи могут легко находить связанные блоки. Область программирования всегда видна, приглашая тем самым учащихся начать программирование. VEXcode VR использует готовых роботов и команды трансмиссии (рис. 1). Это позволяет пользователям управлять своим виртуальным роботом за считанные секунды.

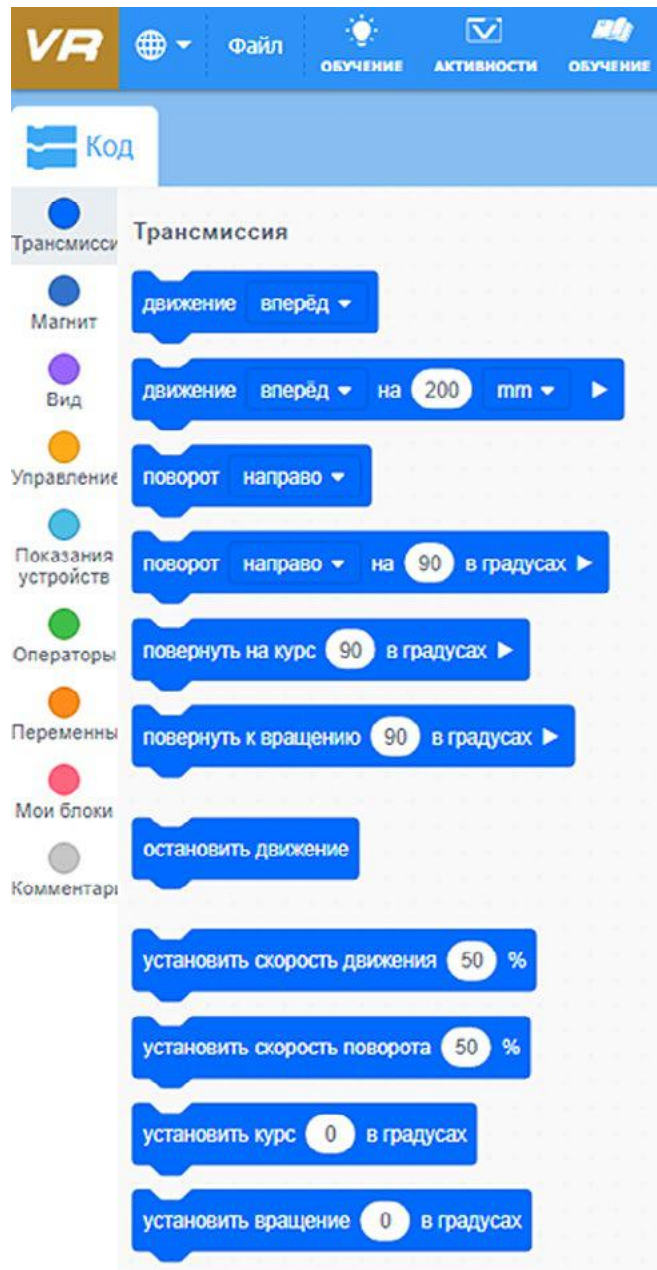
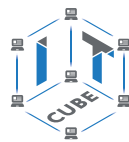


Рис. 1. Блок команд **Трансмиссия**

Робот VEX VR (далее — VR-робот) оснащён датчиками, элементами управления и множеством физических функций. В VEXcode VR представлен только один робот, и он уже заранее настроен. Это устраняет необходимость в настройке конфигурации робота или заранее определённого шаблона проекта.

Мгновенная обратная связь

VEXcode VR стимулирует проведение робототехнических экспериментов и игр. Когда учащиеся запускают проект, они могут сразу увидеть, произвёл ли их робот желаемое действие. Учителя легко контролируют успехи учащихся. Проект в VEXcode VR всегда будет работать одинаково (это не всегда происходит при манипуляциях с физическим роботом). В свою очередь, это позволяет учителям и учащимся сосредоточиться на логике программирования, а не на физике робота или на игровом поле, на котором работает робот.



Игровое поле (площадка) VEXcode VR

VR-робот всегда начинает выполнение программы с одной и той же точки. Учащиеся могут добавлять блоки во время работы своего проекта, останавливать проект в любой момент и перезагружать свою виртуальную площадку одним щелчком мыши. Блоки, не подключённые к основному скрипту, просто игнорируются при запуске проекта.

В VEXcode VR ошибок нет. Учащиеся могут совершать логические ошибки при кодировании, но они не будут разочарованы тем, что их проекты не компилируются и не запускаются.

Способность VEXcode VR обеспечивать мгновенную обратную связь и простота использования позволяют учащимся учиться в процессе написания кода, создавать код с помощью небольших фрагментов, постепенно формируя окончательный вариант программы (скрипта).

Обучение стало наглядным

Окно «Игровое поле» в VEXcode VR (рис. 2) содержит приборную информационную панель, на которой отображаются все данные датчиков от VR-робота (рис. 3). Каждый раз, когда работает VR-робот, учащиеся могут видеть обновление данных датчика в режиме реального времени, получая информацию о том, как эти данные могут быть использованы.

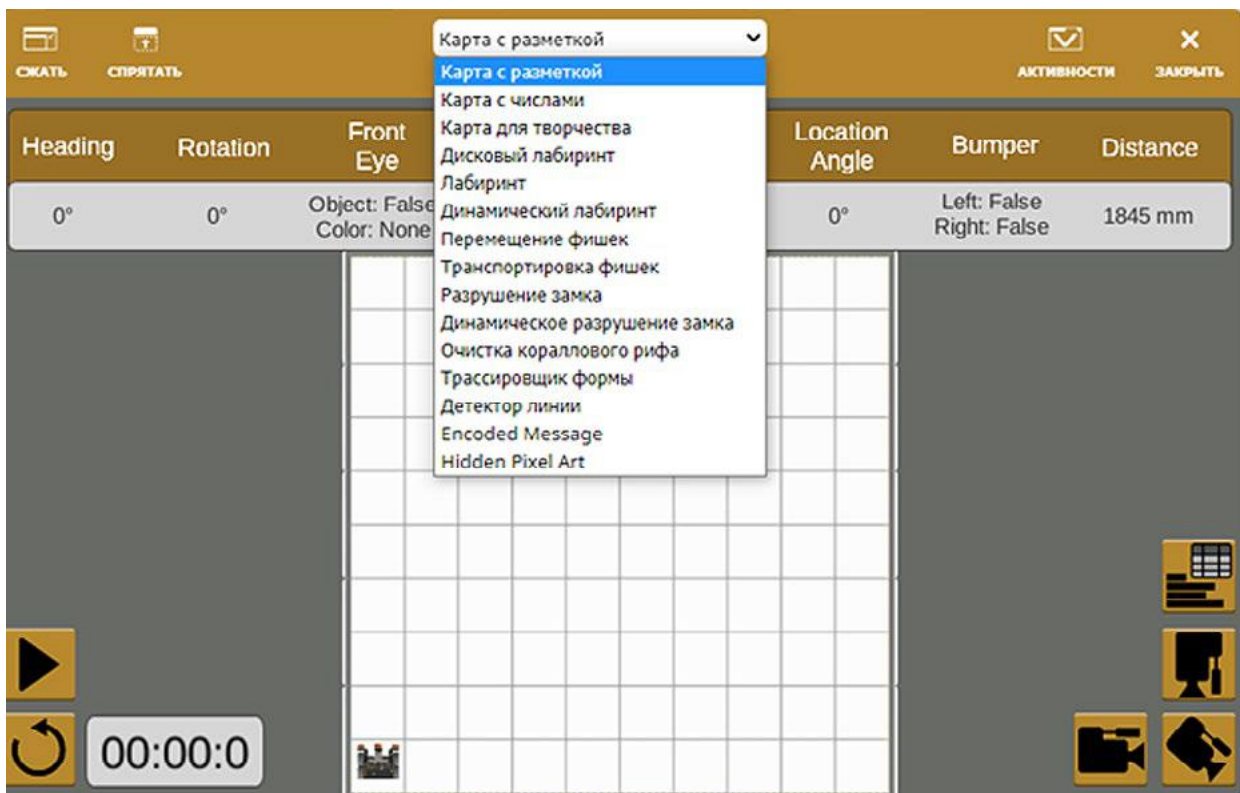


Рис. 2. Игровое поле с выпадающим меню выбора игрового поля



Рис. 3. Приборная панель с показаниями датчиков

VEXcode VR также выделяет зелёным контуром выполняемые блоки в проекте. Эта функция позволяет учащимся наблюдать за ходом выполнения своих проектов. Когда проект VEXcode VR запущен, выполняемый блок окружён светящейся зелёной рамкой. Эта обратная связь позволяет учащимся понять, почему VR-робот выполняет определённый манёвр. Эти данные в реальном времени вместе с выделением выполненных блоков могут помочь обучающимся понять, как робот принимает решение, сделать этот процесс более заметным и конкретным.

VEXcode VR также помогает организовать дистанционную работу. Во всех этих (и не только) ситуациях VEXcode VR может стать отличным дополнением для физических роботов.

Справочник!

Управление VR-роботом

VR-робот имеет следующие элементы управления.

- Трансмиссия с гироскопом: включает категорию блоков **Трансмиссия** в наборе инструментов VEXcode VR.
- Функция «рисование пером», позволяющая размещать перо вверх (чтобы не рисовать) или вниз (для рисования).
- Электромагнит для поднятия дисков с металлическими сердечниками.

Физические характеристики VR-робота

- VR-робот обладает следующими физическими характеристиками.
- Диаметр колёс — 50 мм.
- Колесная база (расстояние между центром переднего колеса и центром заднего колеса) — примерно 50,8 мм.
- Длина VR-робота — 133 мм.

Датчики VR-робота

- VR-робот оснащён следующими датчиками (рис. 4).

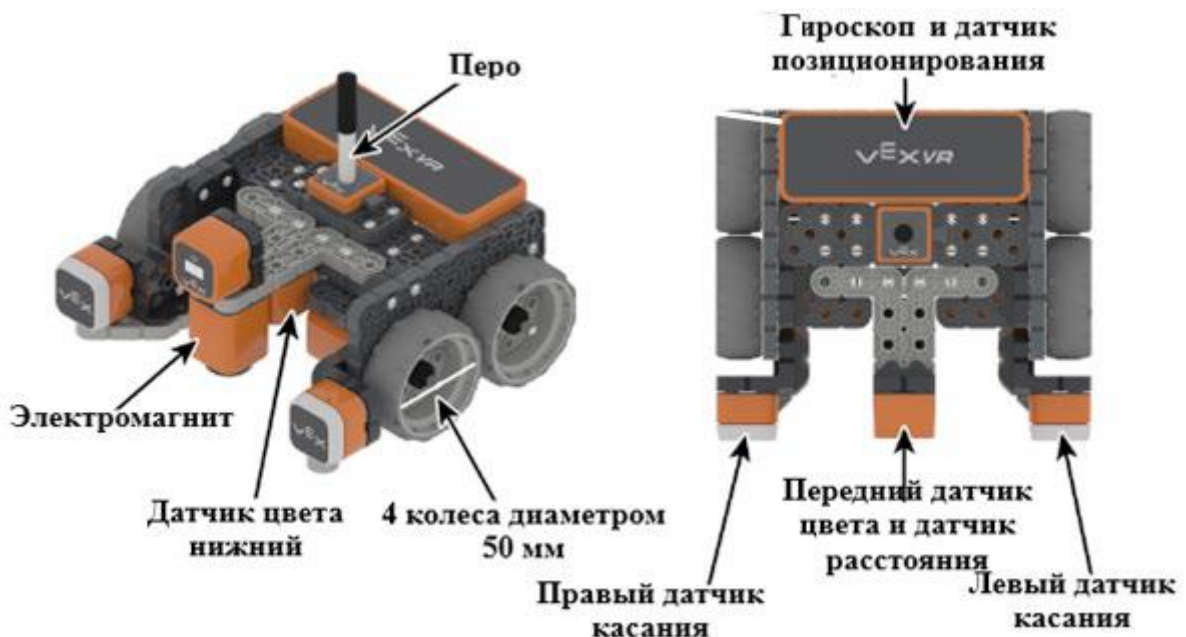
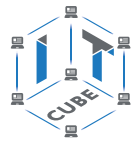


Рис. 4. Конфигурация датчиков VR-робота



- Моторные энкодеры с углом поворота 360° на оборот колеса.
- Передний датчик (сонар) также действует как датчик расстояния и возвращает расстояние до обнаруженного объекта в миллиметрах и дюймах.
- Гироскопический датчик, встроенный в трансмиссию (по часовой стрелке положительный).

Вид сверху на VR-робота (рис. 5).

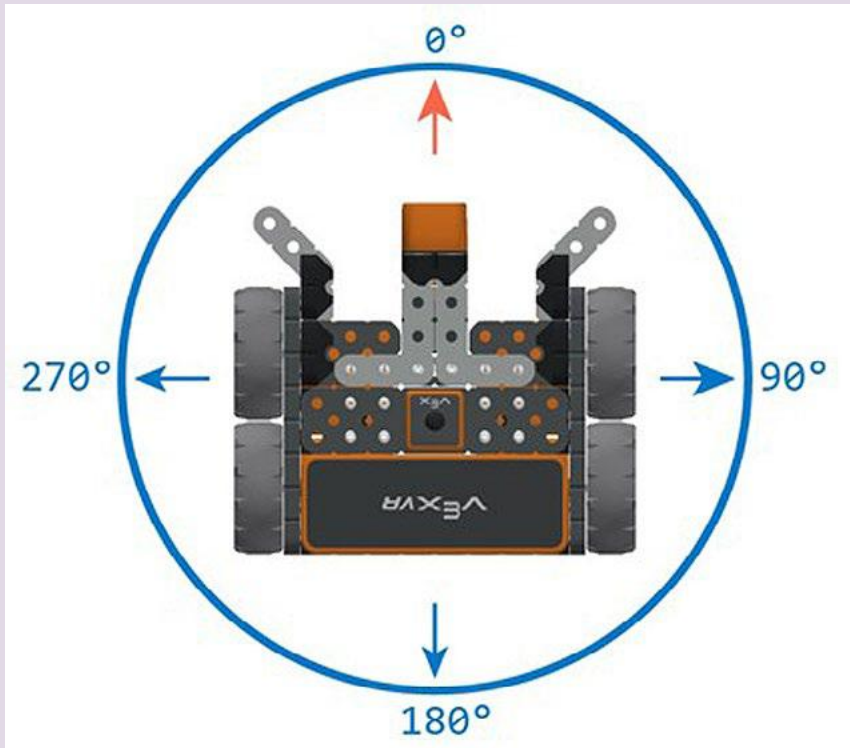


Рис. 5. Направления курса VR-робота, задаваемые блоком **Курс**

- Два датчика цвета: один обращён вперёд, а другой — вниз. Эти датчики могут определять наличие объекта. Если объект есть, датчик также может определять цвет (красный, зелёный, синий, нет).
- Датчик местоположения, который считывает координаты (X, Y) от центральной поворотной точки VR-робота.

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR.

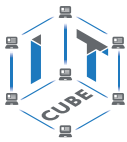
Задание 1. Запрограммировать робота на движение вперёд и назад. Блок программ **Трансмиссия**.

Задание 2. Запрограммировать робота на движение на определённое расстояние. Блок программ **Трансмиссия**.

Задание 3. Запрограммировать робота на повороты в градусах и по курсу. Блок программ **Трансмиссия**.

Задание 4. Проверить загрузку различных игровых полей.

Деятельность учащихся: самостоятельно выполняют предложенные задания.



Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: роботы, датчики, скрипты, игровые поля. Вместе с учащимися обсуждает, как заставить робота двигаться, поворачивать.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт следующие вопросы учащимся.

Для чего нужна программа VEXcode VR?

Что такое датчики и для чего они нужны?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание 1. Зайти на платформу VEXcode VR.

Задание 2. Написать программу движения VR-робота вперёд и назад на 1000 мм (одна клетка поля равна 200 мм).

Задание 3*. Написать программу движения VR-робота по периметру игрового поля (квадрат). Рисовать квадрат во время движения со сторонами разного цвета. Для рисования робот должен выполнить команды **Опустить перо** и **Поднять перо**.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Уроки 2–3

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 1. Знакомство с платформой VEXcode VR.

Тема урока: Создание простейших программ (скриптов), сохранение и загрузка проекта.

Тип урока: комбинированный.

Цель урока: научить учащихся создавать простейшие программы (скрипты) на платформе VEXcode VR.

Время реализации: 2 академических часа.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 + 15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией платформы VEXcode VR (<http://vr.vex.com>).

Примеры различных программ, созданных с помощью платформы, можно найти в меню **Файл** → **Открыть примеры** (рис. 6).

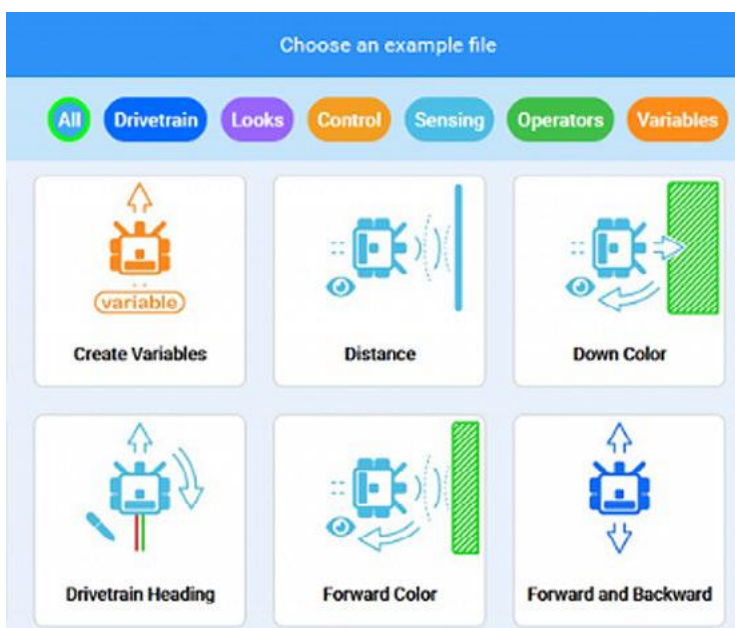


Рис. 6. Примеры программ, созданных на платформе VEXcode VR

Создайте простую программу движения робота по периметру игрового поля **Карта с разметкой** (рис. 7). Размеры одной клетки поля составляют 200 × 200 мм.

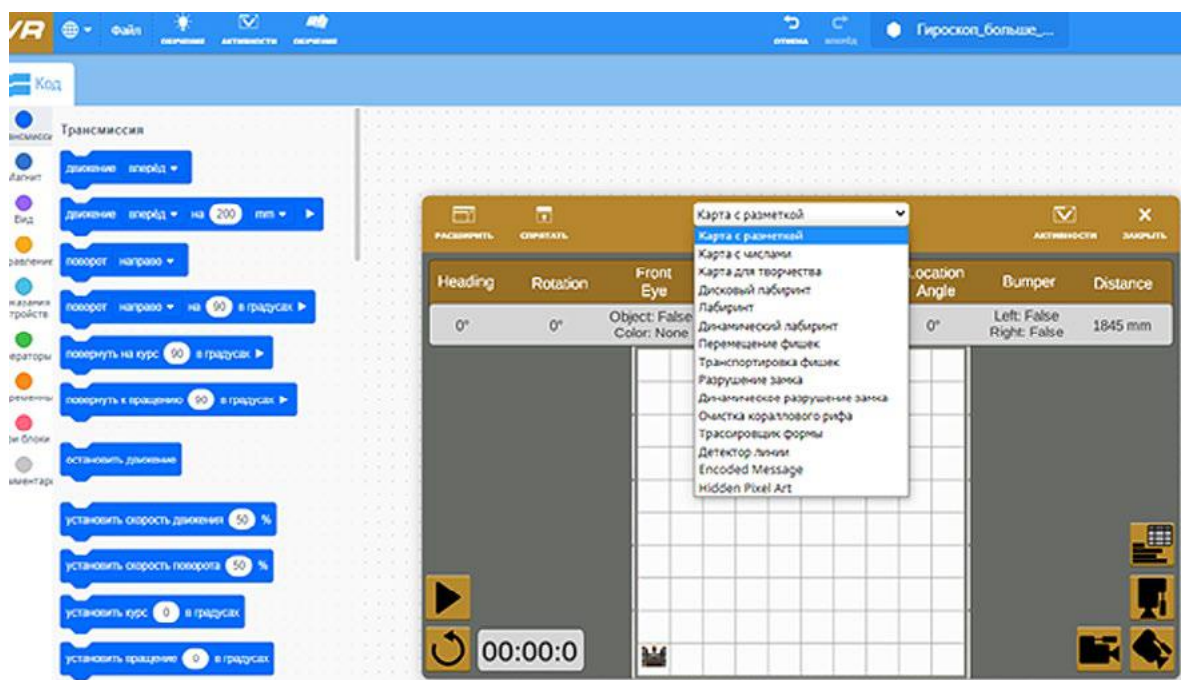


Рис. 7. Игровое поле **Карта с разметкой**

Простейшая программа будет выглядеть следующим образом (рис. 8).

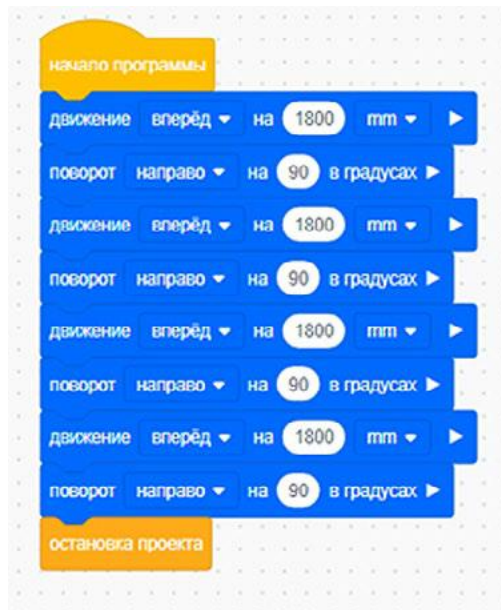


Рис. 8. Скрипт обхода VR-роботом игрового поля по периметру

Можно немного усложнить программу. Пусть VR-робот при обходе поля рисует линию. Для этого перейдём в блок программ **Вид** и перед началом движения поставим два блока, показанные на рисунке 9.

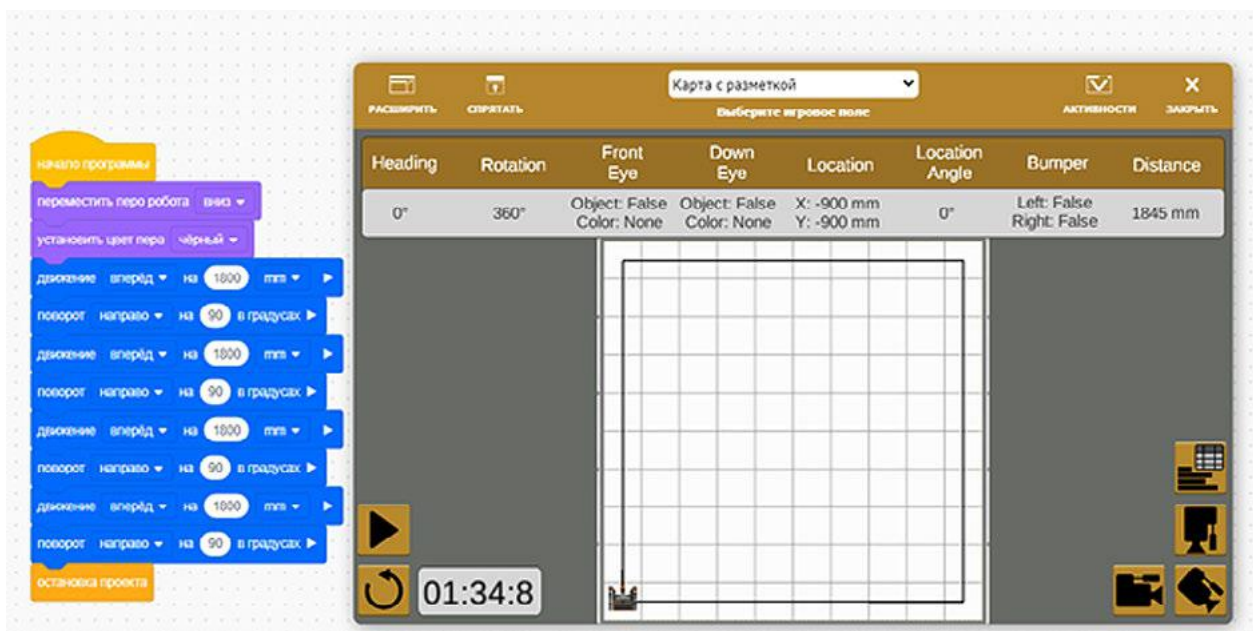


Рис. 9. Скрипт рисования VR-роботом квадрата

Созданную программу необходимо сохранить на своём компьютере или планшете. Для этого нужно кликнуть мышкой на верхней панели и ввести имя файла (рис. 10).

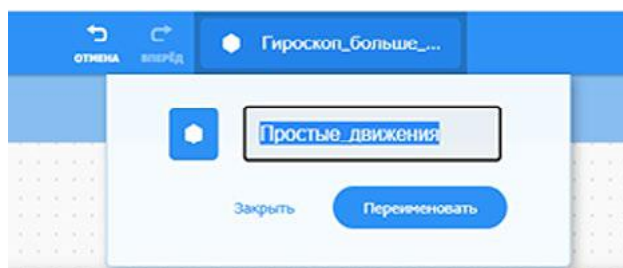
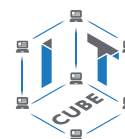


Рис. 10. Переименование рабочего файла

Сохранить этот файл на своё устройство можно через меню **Файл** → **Сохранить на ваше устройство**, файл будет скачан в папку **Загрузки** на вашем устройстве.

Рассмотрим теперь позиционирование VR-робота на игровой площадке с помощью датчика положения (гироскопа). Игровая площадка разделена на координатные плоскости (декартовы координаты), как показано на рисунке 11.

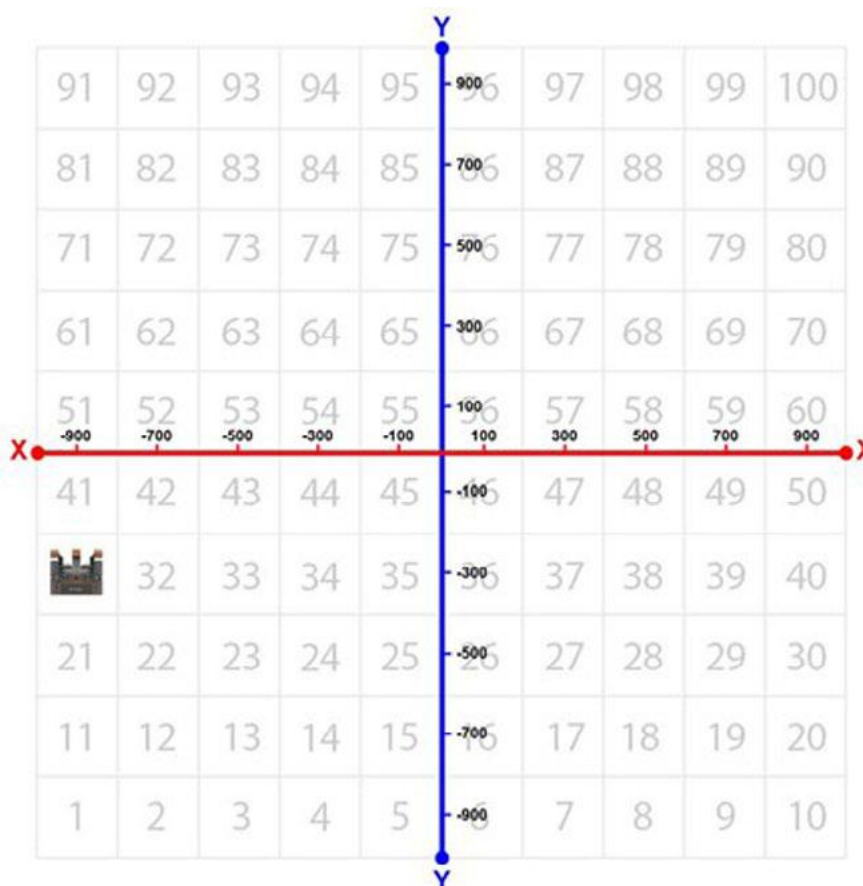


Рис. 11. Координатная плоскость игрового поля

Движение VR-робота по координатной плоскости задаётся следующим образом (рис. 12).

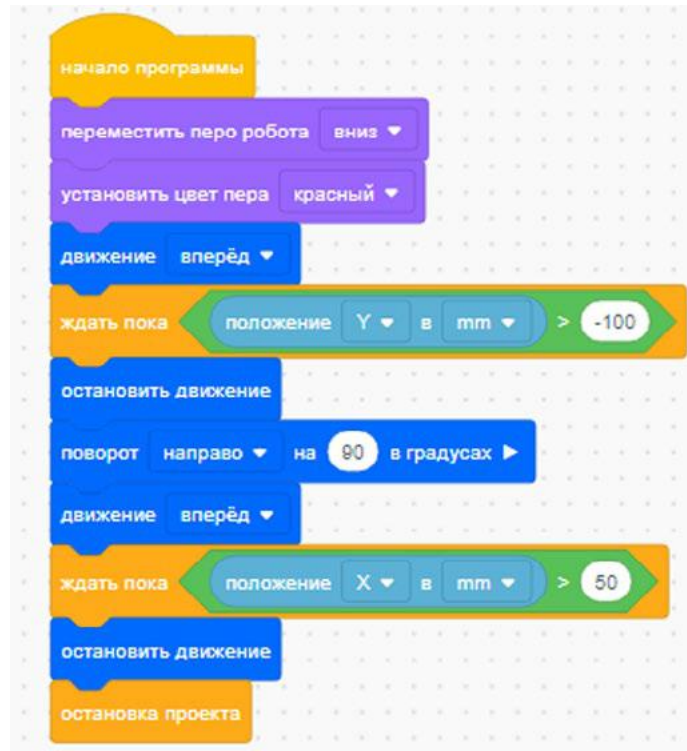


Рис. 12. Перемещение VR-робота по координатам

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 + 15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR, используя знания, полученные в ходе изложения нового материала.

Задание 1. Напишите скрипт рисования VR-роботом квадрата со сторонами разного цвета.

Задание 2. Напишите скрипт рисования VR-роботом ромба со сторонами разного цвета, используя гироскоп.

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Повторение нового материала (5 + 5 мин)

Деятельность учителя: кратко повторяет новый материал: программирование простых движений робота, датчики, скрипты, игровые поля. Вместе с учащимися обсуждает, как заставить робота двигаться, поворачивать, рисовать.

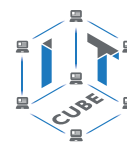
Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 + 5 мин)

Деятельность учителя: задаёт следующие вопросы учащимся.

- Для чего нужна программа **Трансмиссия**?
- Какой блок программ отвечает за движение робота?
- Как регулировать скорость движения робота?
- Как работает блок программ **Курс**?
- Как направить движение робота по диагонали игрового поля?
- Для чего используется магнит? Как им управлять?

Деятельность учащихся: отвечают на предложенные вопросы.

**Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 + 3 мин)**

Деятельность учителя: сообщает учащимся домашнее задание.

Задание 1. Написать программу рисования VR-роботом инициалов учащегося на игровом поле.

Задание 2. Написать программу движения VR-робота курсом 45, 135, 200, 300 градусов на расстояние 1000 мм и объяснить работу программы учителю.

Задание 3*. Написать программу движения VR-робота из точки с координатами (–900; –900) в точку с координатами (–900; –100).

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

VI. Этап рефлексии деятельности на уроке (2 + 2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятым.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXcode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Уроки 4–5

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 2. Программирование робота на платформе.

Тема урока: Математические и логические операторы, блоки вывода информации в окно вывода, блоки трансмиссии.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с новой группой блоков управления VR-роботом.

Время реализации: 2 академических часа.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока**Этап I. Постановка цели и задач урока, мотивация к учебной деятельности и изложение нового материала (15 + 15 мин)**

Деятельность учителя: объясняет новый материал с демонстрацией создания скрипта управления роботом на платформе VEXcode VR (<http://vr.vex.com>).

Группа арифметических операторов располагается в зоне зелёной кнопки (рис. 13).

Демонстрирует учащимся, как работает математический оператор (рис. 14).

Алгоритм этого скрипта выглядит следующим образом: **Очистить** все строки (очистка значений предыдущих программ на **Экране**) → **Переместить перо вниз** (активируем перо робота для рисования) → **Движение вперёд на** (здесь задаётся не фиксированное значение, а вычисленное $100 * 10 = 1000$ мм с помощью указанного оператора) → **Повернуть на 135 град** → **Движение вперёд на** (здесь также вычисляется значение, при этом нужно учитывать правила последовательности выполнения математических операций, сначала выполняем умножение, а затем — сложение) → **Поворот налево** → **Движение вперёд на** (случайно сгенерированное значение от 500 до 600 мм) → **Вывод на печать сообщения в Экран**.

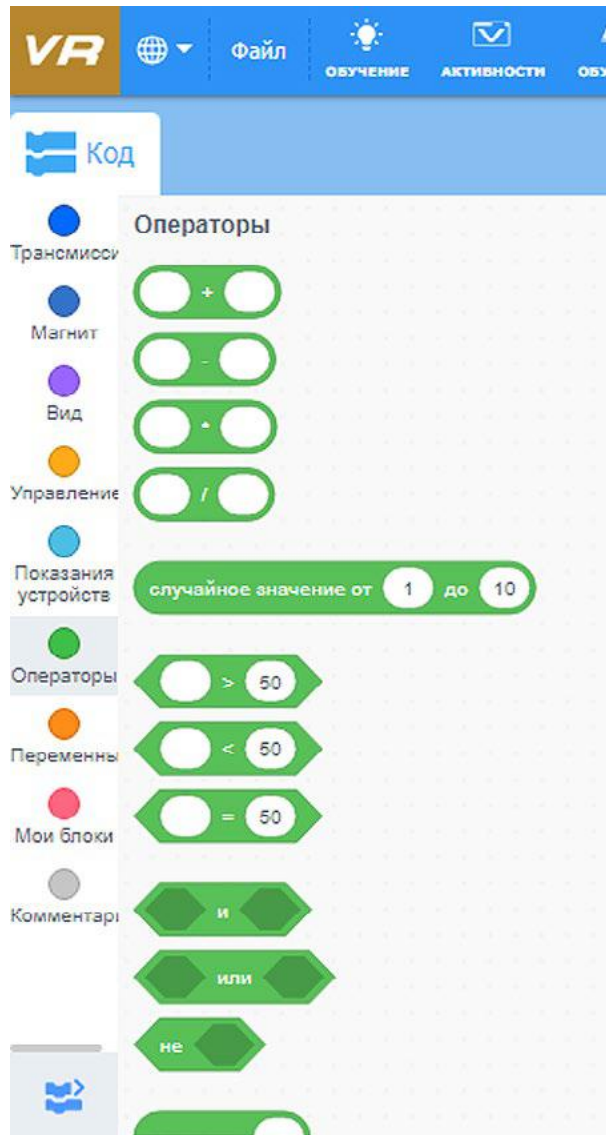
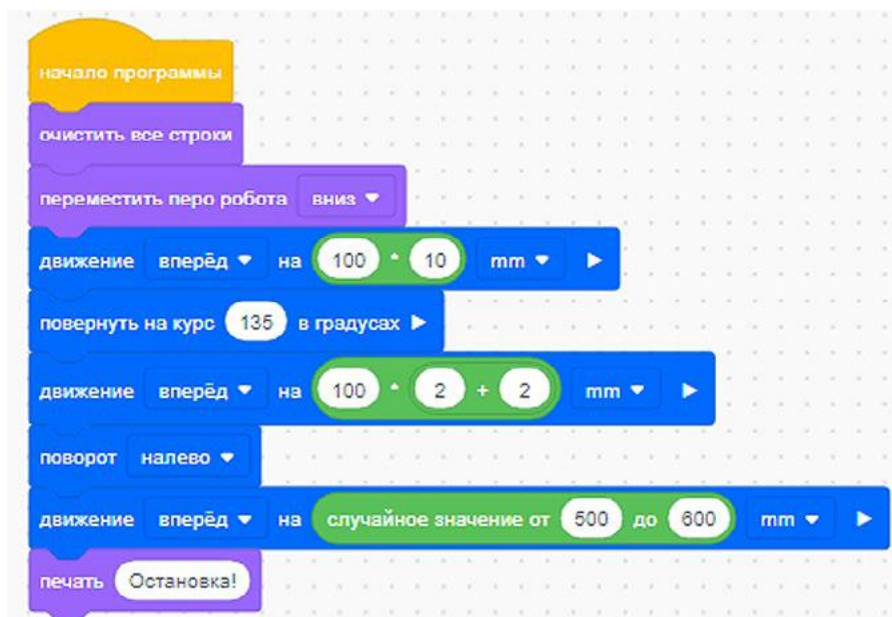


Рис. 13. Группа математических и логических операторов



Ниже приведены рисунки 15 и 16, поясняющие некоторые действия алгоритма данного скрипта.

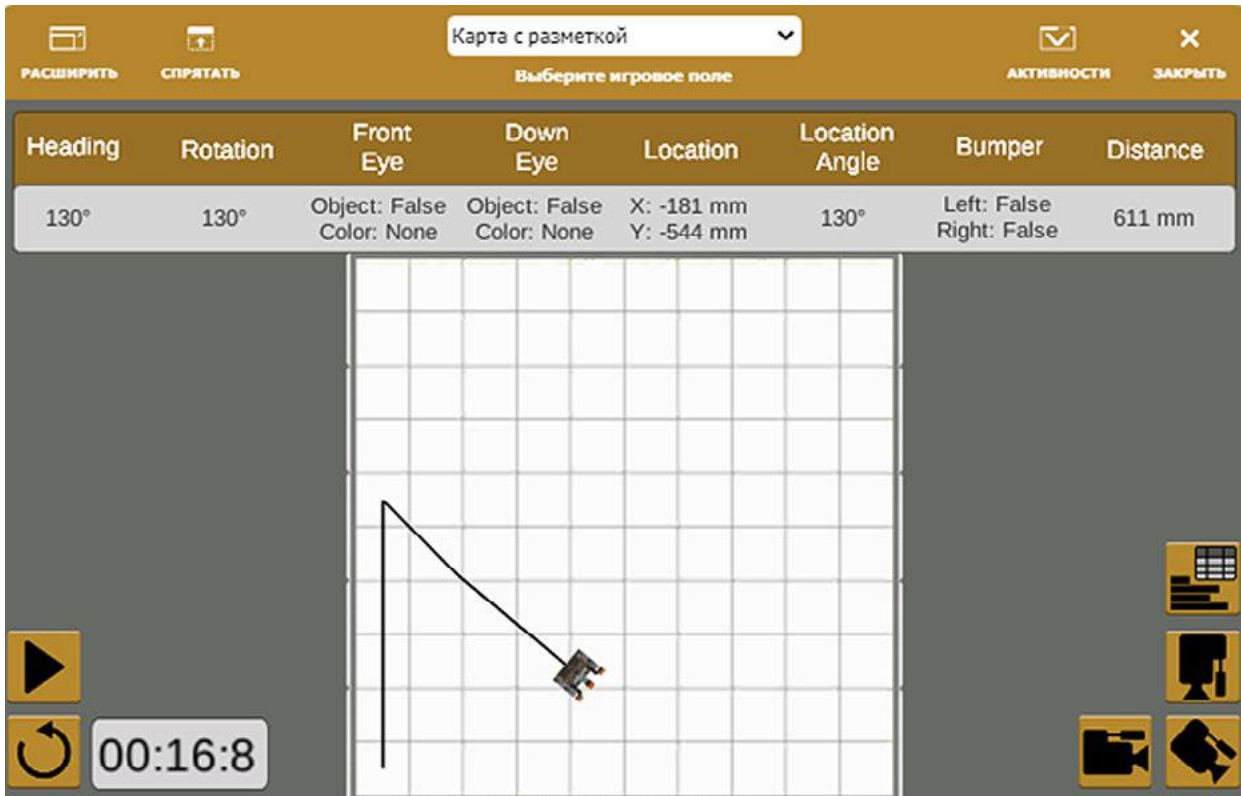


Рис. 15. Траектория движения VR-робота

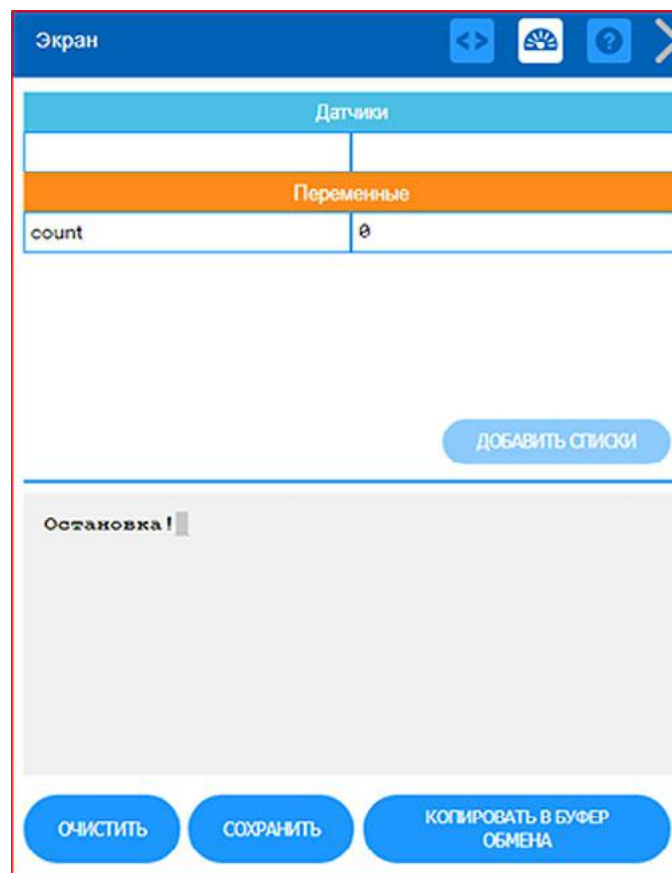
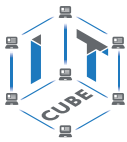


Рис. 16. Вывод сообщения на печать в Экран



Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 + 15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR, используя знания, полученные в ходе изложения нового материала.

Задание 1. Воспроизвести скрипт, описанный выше, на компьютерах учащихся.

Задание 2. Изменить некоторые параметры скрипта и исследовать, какие изменения при этом произойдут в движении VR-робота.

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Повторение нового материала (5 + 5 мин)

Деятельность учителя: кратко повторяет новый материал: программирование математических операций, порядок выполнения математических операций, вывод печати на **Экран**, блоки управления.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 + 5 мин)

- Деятельность учителя: задаёт учащимся следующие вопросы.
- Для чего нужен блок программ **Операторы**?
- Какой блок программ отвечает за движение робота?
- Для чего нужны логические операторы?
- Как задать случайное число?
- Как задать различные функции (sin, cos, степень и т. д.)?
- Какие ещё функции можно задать с помощью математических операторов?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 + 3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание 1. Написать произвольную программу с использованием всех арифметических операторов (сумма, произведение, разность, деление).

Задание 2. Написать программу движения VR-робота из точки (−900; −900) в точку (100; 100) и объяснить работу программы учителю.

Задание 3*. Написать произвольную программу движения VR-робота с использованием других математических функций.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 + 2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

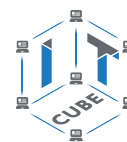
Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).



Уроки 6–7

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 2. Программирование робота на платформе.

Тема урока: Блоки управления, блоки переменных, блоки датчиков, блоки вида, магнит.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с новой группой блоков управления VR-роботом и возможностями программирования с их помощью.

Время реализации: 2 академических часа.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности и изложение нового материала (15 + 15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией создания скрипта управления роботом на платформе VEXcode VR (<http://vr.vex.com>).

Рассмотрим панель блоков управления с указанием их функционала (таблица 1).

Таблица 1

Блоки управления и их функционал

 Трансмиссия	 Магнит	 Вид	 Управление	 Показания устройств	 Операторы	 Переменные	 Мои блоки
Блоки, управляющие движениями робота	Блоки, управляющие активацией магнита	Блоки, управляющие печатью в консоли и пером	Блоки условных переходов и циклов	Блоки, управляющие датчиками	Блоки математических операторов	Блоки создания и управления переменными	Создание собственных блоков

Основное достоинство этих блоков — возможность создать условный переход (цикл) с их помощью. Важной группой блоков является группа **Управление**, с помощью которых организуются циклы и переходы по условиям. Виды блоков условных переходов и их функции приведены в таблице 2.

Таблица 2

Блоки условных переходов

Блок условного перехода	Функция
Повторять N раз	Устанавливает количество повторений фрагмента программы
Всегда	Организует бесконечное повторение фрагмента программы
Если...тогда	Переход по условию
Если...тогда...иначе...	Переход по условиям
Ждать пока...	Ожидание по выполнению условий
Повторять пока...	Повторение до выполнения условий

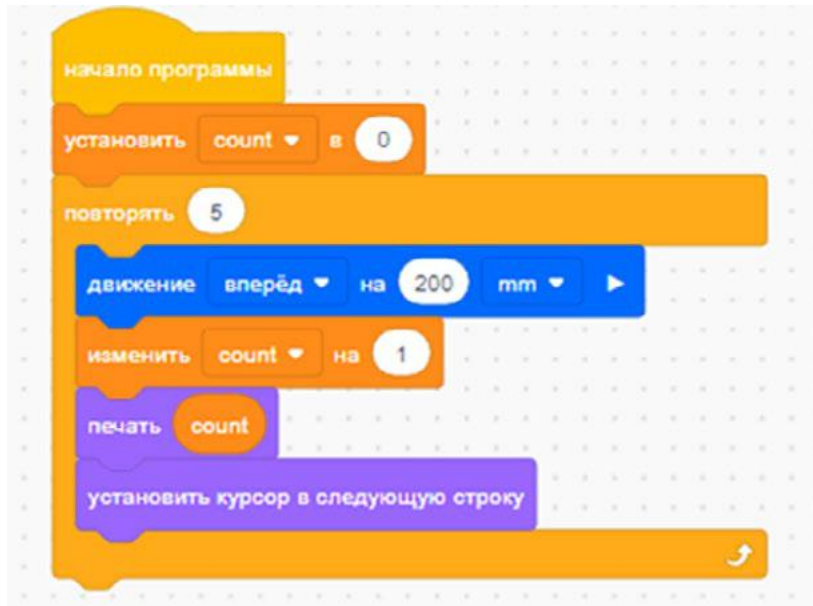


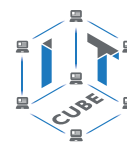
Рис. 17. Скрипт с использованием цикла **Повторять** и выводом на консоль **Экрана** значения переменной **COUNT** (счётчика итераций)

Учитель приводит примеры работы каждого блока. Например, блок, приведённый на рисунке 17, реализует следующий алгоритм: созданной обучающимся переменной **COUNT** присваивается значение 0 (чтобы реализовать счётчик итераций) → выставляем блок **Повторять 5 раз** и внутри этого блока помещаем другие блоки: **Движение вперёд на 200 мм** → **Изменить COUNT на 1** → вывести текущее значение переменной на **Экран** (рис. 18).



Рис. 18. Выведенные на **Экран** значения переменной **COUNT**

Ещё один из важных блоков — это блок **Показания устройств (Датчики)**. Описание функционала датчиков приведено в таблице 3.



Справочник!

Таблица 3

Датчики. Описание функционала

Блок	Функция
Трансмиссия	4 датчика движения, которые помогают сформулировать некоторые условия для управления роботом
Бампер	При касании одним из датчиков (левым или правым) препятствия возвращают значение ИСТИНА или ЛОЖЬ. Рекомендуется разобрать с учащимися принцип действия датчика касания
Датчик расстояния	Возвращает расстояние до объекта (препятствия) или сигнализирует об обнаружении объекта. Работает как сонар (ультразвуковой излучатель / приёмник). Рекомендуется разобрать с учащимися принцип действия датчика расстояния
Датчики цвета	Определяют цвет (синий, зелёный, красный, никакой) внизу робота и перед роботом. Рекомендуется разобрать с учащимися принцип действия датчиков цвета
Положение на поле	Определяет координаты робота (X, Y) на игровом поле, а также текущий угол в градусах. Рекомендуется разобрать с учащимися принцип действия датчика положения на поле

Простейшая программа, иллюстрирующая работу управляющих блоков и датчиков VR-робота, — это уже знакомый нам обход игрового поля по периметру (рис. 19).

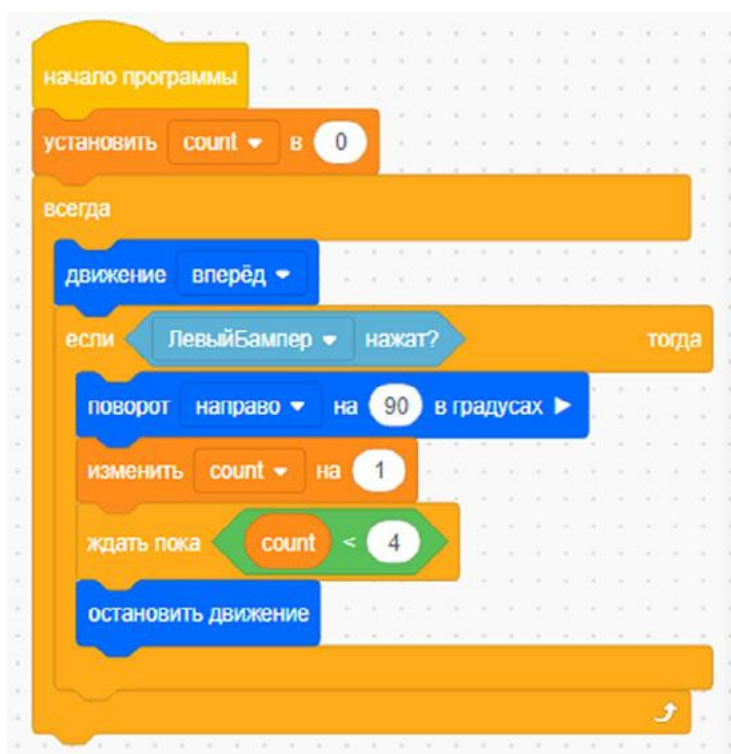


Рис. 19. Обход VR-роботом периметра игрового поля с использованием датчика касания

Алгоритм работает следующим образом: присвоим переменной **COUNT** (счётчик) значение 0 → используем управляющий блок **Всегда** (бесконечный цикл повторения), внутри которого разместим основной фрагмент программы — двигаемся вперёд, пока датчик не коснётся стены, — сработал датчик — поворачиваем вправо и снова двигаемся вперёд, при этом увеличивая значение переменной **COUNT** на единицу → повторяем цикл до тех пор, пока выполняется условие **COUNT < 4** (робот обходит 4 стены периметра), как только условие перестаёт выполняться, программа останавливает робота.

Более сложный алгоритм обхода VR-роботом периметра реализован на рисунке 20. В нём используется датчик расстояния и остановки проекта при его завершении. Также здесь приведён блок **Если...тогда...иначе** для организации условного перехода.



Рис. 20. Обход игрового поля по периметру с использованием датчика расстояния

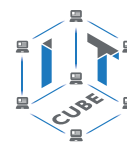
Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 + 15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR, используя знания, полученные в ходе изложения нового материала.

Задание 1. Воспроизвести скрипт, описанный выше, на компьютерах учащихся.

Задание 2. Изменить некоторые параметры скрипта и исследовать, какие изменения при этом произойдут в движении VR-робота, например изменить условие **COUNT < 4**.



Задание 3. Запрограммировать VR-робота, чтобы он чертил красную линию по ходу своего движения.

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Повторение нового материала (5 + 5 мин)

Деятельность учителя: кратко повторяет новый материал: блоки управления, задание условных переходов, организация цикла, работа датчиков.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 + 5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

Для чего нужен блок программ **Управление**?

Какой блок программ отвечает за движение робота?

Для чего нужны датчики и какие они бывают?

Как остановить проект?

Что такое вложенный цикл?

Что такое условный переход и как его организовать?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 + 3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание*. Написать программу движения VR-робота по периметру игрового поля, рисующего по ходу движения линии разных цветов, с использованием цикла.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 + 2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Уроки 8–9

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 3. Датчики и обратная связь.

Тема урока: Датчик местоположения, датчик направления движения.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с новой группой блоков управления VR-роботом и возможностями программирования с их помощью.

Время реализации: 2 академических часа.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности и изложение нового материала (15 + 15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией создания скрипта управления роботом на платформе VEXcode VR (<http://vr.vex.com>).

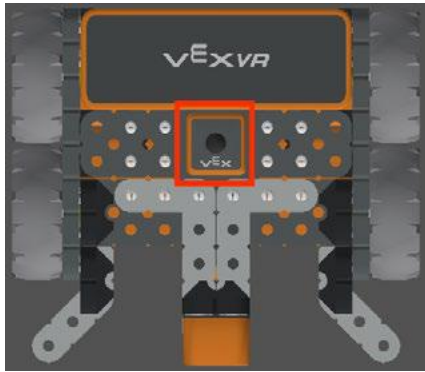


Рис. 21. Датчик местоположения VR-робота

Датчик местоположения

VR-робот имеет встроенный датчик местоположения, который сообщает его положение (X, Y). Местоположение VR-робота определяется точкой поворота спереди в центре (рис. 21). Показано также местоположение ручки на VR-роботе. Эта информация важна для понимания того, какой точкой позиционируется VR-робот на игровом поле. Значения датчика местоположения отображаются на панели управления в VEXcode VR (рис. 22).

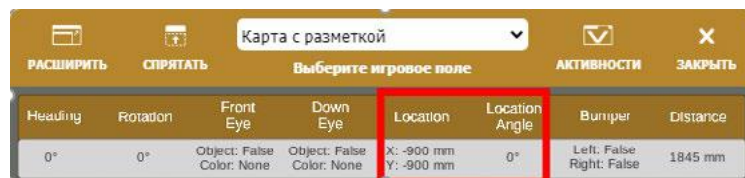


Рис. 22. Показания датчика местоположения на панели управления

Продемонстрируем скрипт достижения VR-роботом координат $Y > 0$ (рис. 23) и его положение после выполнения данного скрипта (рис. 24).

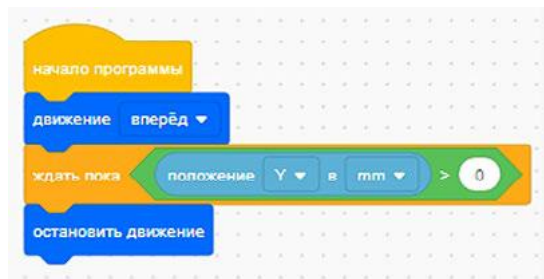


Рис. 23. Скрипт достижения VR-роботом координат $Y > 0$

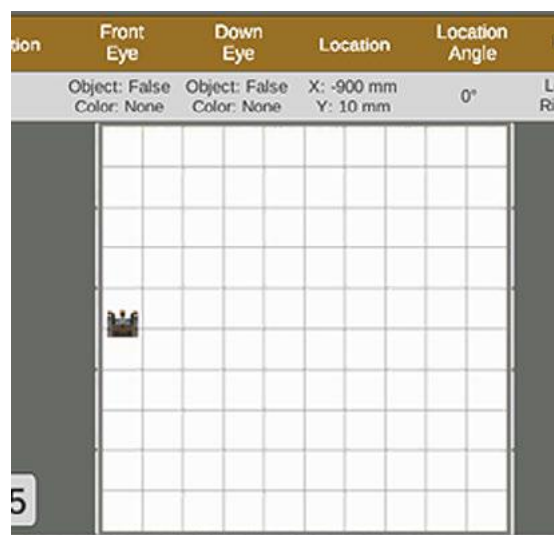
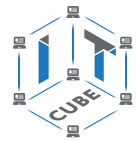


Рис. 24. Положение VR-робота после выполнения скрипта



Датчики	
Координата X в мм	-900
Координата Y в мм	10
Переменные	
count	0

Рис. 25. Отображение координат VR-робота на консоли Экрана

Значения датчика местоположения могут отображаться на консоли Экрана в VEXcode VR (рис. 25).

Блоки из категории **Положение на поле** (рис. 26) в панели инструментов можно добавить и удалить из консоли Экрана, установив или сняв соответствующий флажок.

Переменные можно добавлять и удалять из консоли Экрана, устанавливая и снимая флажок в соответствующем поле. В VEXcode VR перечень переменных всегда начинается с переменной **myVariable**. Создать список можно в разделе **Переменные**, а затем отобразить его через консоль Экрана, нажав кнопку **Добавить списки**.

Мониторинг значений датчиков в консоли монитора позволяет пользователю просматривать данные датчиков в режиме реального времени. Консоль Экрана также можно использовать для отслеживания значений переменных. Консоль монитора может предоставлять отчёты в реальном времени по определённой переменной в проекте.

Также можно отобразить на консоли Экрана и на контрольной панели угол положения робота в градусах (рис. 27).

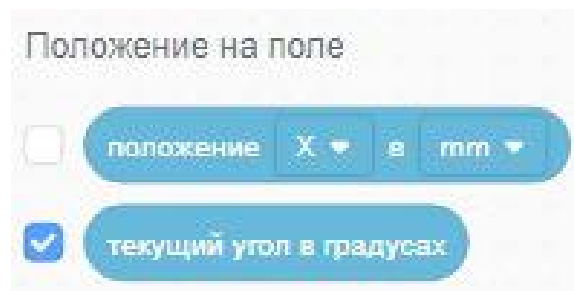


Рис. 26. Блоки из категории **Положение на поле**

Датчики	
Текущий угол в градусах	0
Переменные	
count	0

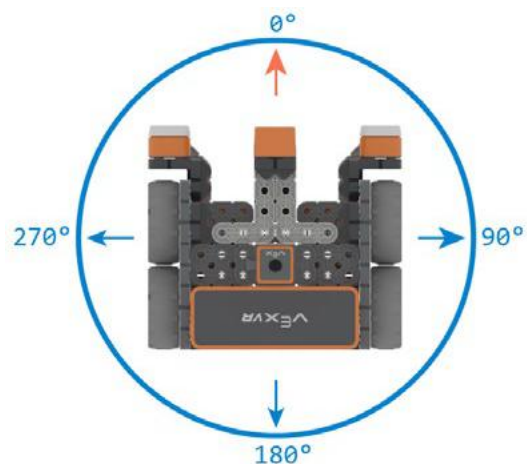


Рис. 27. Позиционирование текущего угла VR-робота

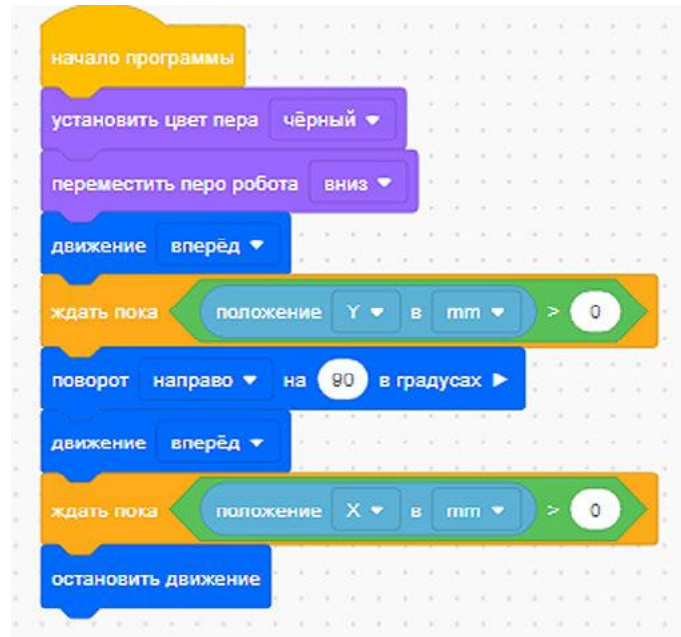
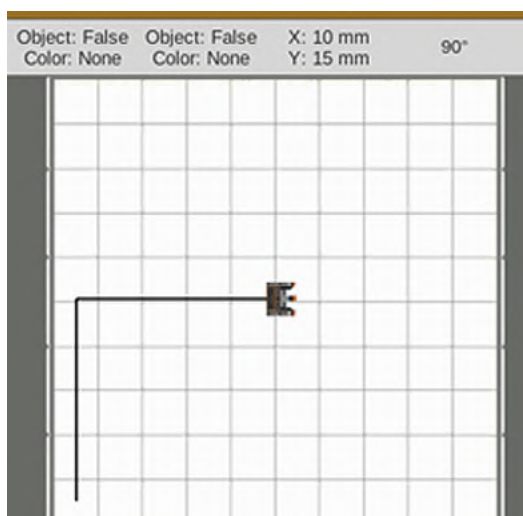


Рис. 28. Скрипт проекта с датчиком местоположения

Пример проекта с использованием датчика местоположения приведён на рисунке 28. На рисунке 29 показано отражение параметров этого проекта с датчиком местоположения.



Датчики	
Координата X в мм	10
Координата Y в мм	15
Переменные	
count	0

Рис. 29. Отражение параметров проекта с датчиком местоположения

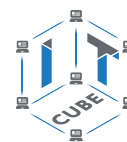
Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 + 15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR, используя знания, полученные в ходе изложения нового материала.

Задание 1. Воспроизвести скрипты, описанные выше, на компьютерах учащихся.

Задание 2. Изменить параметры (X, Y) скрипта и исследовать, какие изменения при этом произойдут в движении VR-робота.



Задание 3. Запрограммировать VR-робота с использованием датчика местоположения, чтобы он начертил домик по ходу своего движения.

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Повторение нового материала (5 + 5 мин)

Деятельность учителя: кратко повторяет новый материал: датчик местоположения, задание координат местоположения, отражение координат на панели управления и консоли **Экрана**.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 + 5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Для чего нужен датчик местоположения?
- Как можно контролировать показания датчика?
- Как задать курс роботу?
- Как вывести данные на консоль **Экрана**?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 + 3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Написать программу движения VR-робота для всех фигур на игровом поле

Трассировщик формы.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 + 2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).
2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).
3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Уроки 10–11

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 3. Датчики и обратная связь.

Тема урока: Датчики цвета. Дисконный лабиринт.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с новой группой блоков управления VR-роботом и возможностями программирования с их помощью.

Время реализации: 2 академических часа.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности и изложение нового материала (15 + 15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией создания скрипта управления роботом на платформе VEXcode VR (<http://vr.vex.com>).

Датчики цвета являются одними из основных датчиков в задачах программирования роботов. Эти датчики используются для организации условных переходов и очень часто для задания траектории движения робота (движения по чёрной линии).

В VR-робота встроены 2 датчика цвета (рис. 30): один направлен вниз, а другой — вперёд.



Рис. 30. Датчики цвета на VR-роботе

Учащимся совместно с учителем предлагается написать скрипт движения VR-робота по дисковому лабиринту (рис. 31).

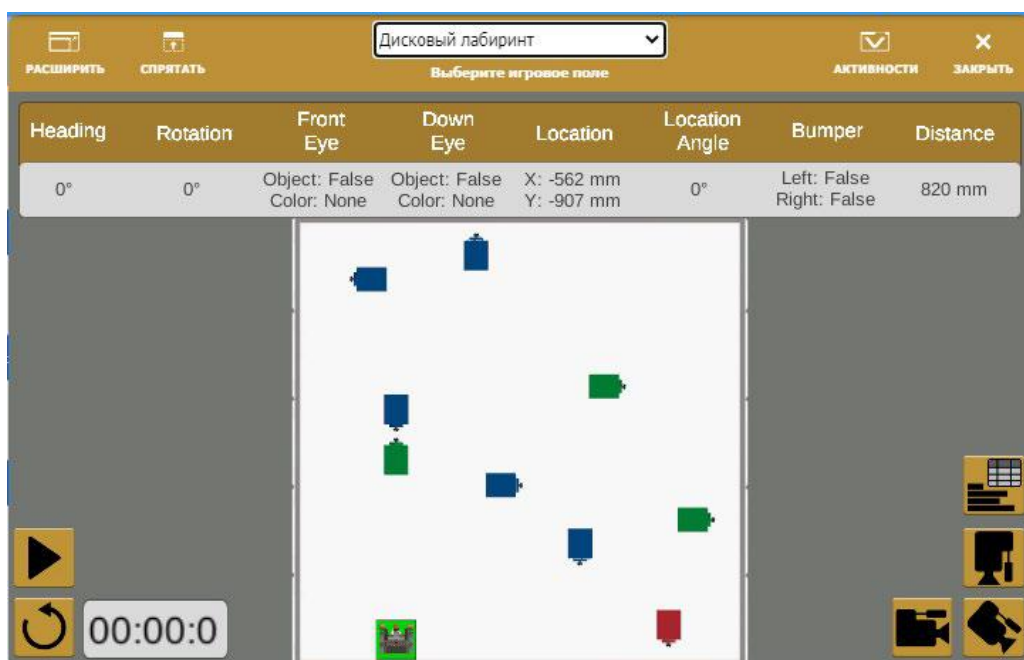
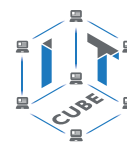


Рис. 31. Игровое поле Дисковый лабиринт



Учитель предлагает учащимся сформулировать алгоритм движения робота по этому лабиринту. Очевидно, что он будет выглядеть следующим образом: **Старт** (движение вперёд) → **Если** диск зелёного цвета, **то** повернуть направо на 90 градусов → **Если** диск синего цвета, **то** повернуть налево → **Если** диск красного цвета, **то** остановить движение.

Готовый скрипт будет выглядеть следующим образом (рис. 32).

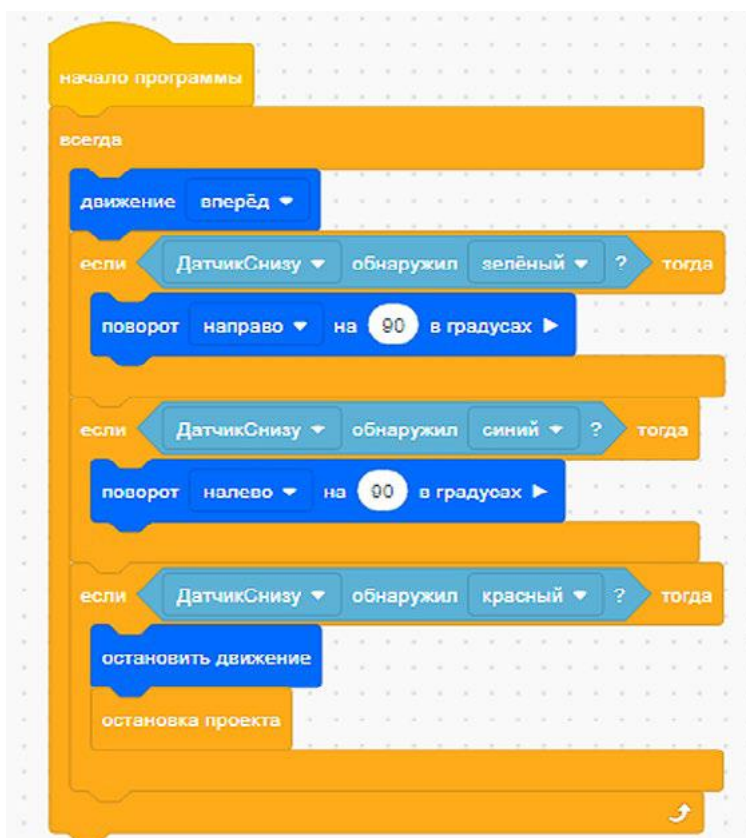


Рис. 32. Скрипт движения VR-робота по дисковому лабиринту

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 + 15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR, используя знания, полученные в ходе изложения нового материала.

Задание 1. Создать скрипт прохождения VR-роботом дискового лабиринта с использованием датчика цвета сверху.

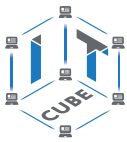
Задание 2. Организовать соревнование «Чей VR-робот быстрее всех пройдёт дисковый лабиринт?» (используем таймер).

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Повторение нового материала (5 + 5 мин)

Деятельность учителя: кратко повторяет новый материал: датчики цвета (верхний и нижний), движение по дисковому лабиринту, отражение данных на панели управления и консоли **Экрана**.

Деятельность учащихся: демонстрируют, как усвоили материал.



Этап IV. Проверка понимания и первичное закрепление (5 + 5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Для чего нужен датчик цвета?
- Как можно контролировать показания датчика?
- Как задать параметры датчика цвета?
- Как вывести данные на консоль **Экрана**?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 + 3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Написать программу движения VR-робота по дисковому лабиринту без использования блока **Если...то**.

Важно!

Использовать блок **Ждать пока**.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 + 2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Уроки 12–15

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 3. Датчики и обратная связь.

Тема урока: Датчик расстояния. Простой лабиринт. Динамический лабиринт.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с новой группой блоков управления роботом и возможностями программирования с их помощью.

Время реализации: 4 академических часа.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности и изложение нового материала (15 + 15 + 15 + 15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией создания скрипта управления роботом на платформе VEXcode VR (<http://vr.vex.com>).

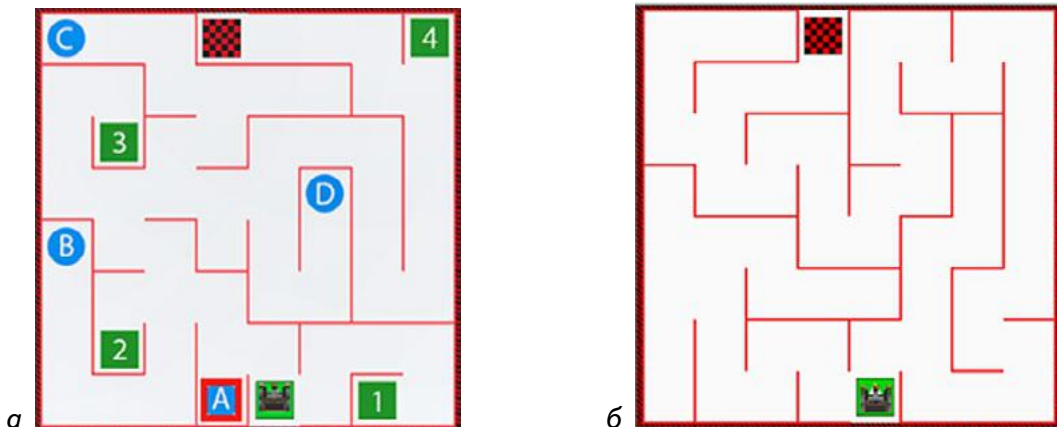
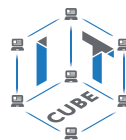


Рис. 33. Простой (а) и динамический (б) лабиринты

Прохождение по лабиринту целесообразно выполнять с использованием различных датчиков: касания, расстояния, цвета, местоположения. При изучении нового материала рекомендуется разбить весь проход лабиринта на фрагменты, чтобы показать учащимся оптимальные режимы прохождения лабиринта VR-роботом.

В игровых полях платформы VEXcode VR представлены 2 типа лабиринтов — простой (статический) и динамический (рис. 33).

Динамический лабиринт — это лабиринт, в котором его конфигурация изменяется при каждой новой загрузке скрипта. В **простом лабиринте** удобно обрабатывать фрагменты движения VR-робота, например поставить ему задачу: «Добраться в точку А со старта». Пример прохождения VR-роботом простого лабиринта от старта до точки В приведён на рисунке 34.

Рассмотрим более сложную задачу: «Как довести робота до цифры 4 за минимально короткое время в простом лабиринте?» В данном скрипте используются: условный переход **Ждать пока** и два датчика — **ЛевыйБампер** (датчик касания) и **Датчик расстояния**. Вместо датчика расстояния можно также использовать датчик касания, но тогда увеличится время прохождения лабиринта на лишние манёвры робота.

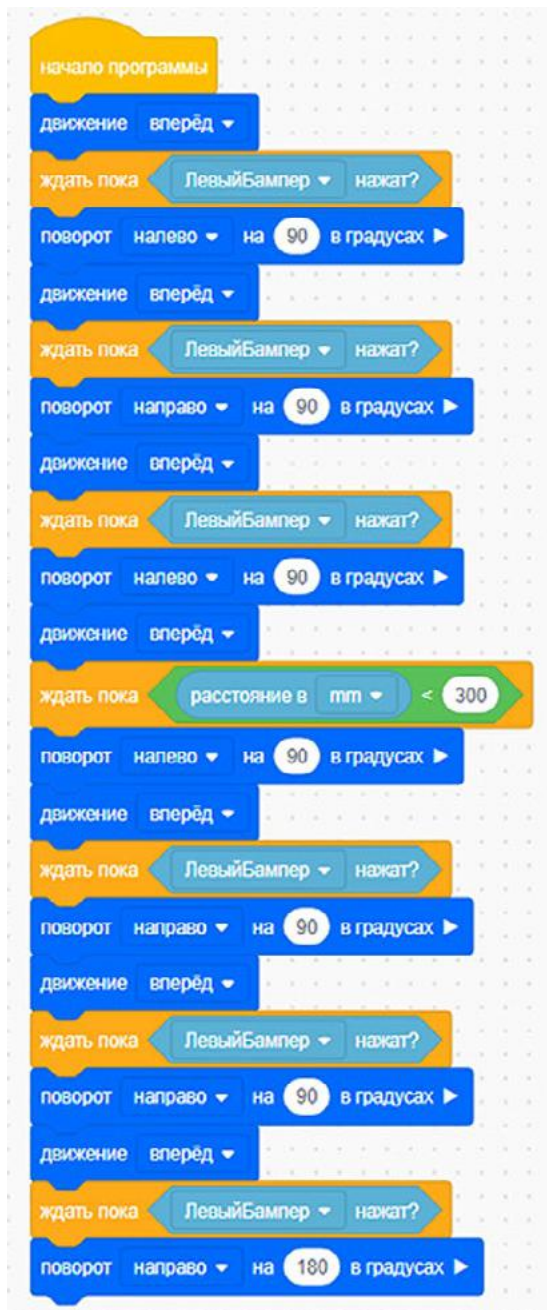


Рис. 34. Прохождение VR-роботом простого лабиринта до точки В

Рассмотрим универсальную программу, которая подходит для прохождения VR-роботом как динамического, так и для статического лабиринта (рис. 35).

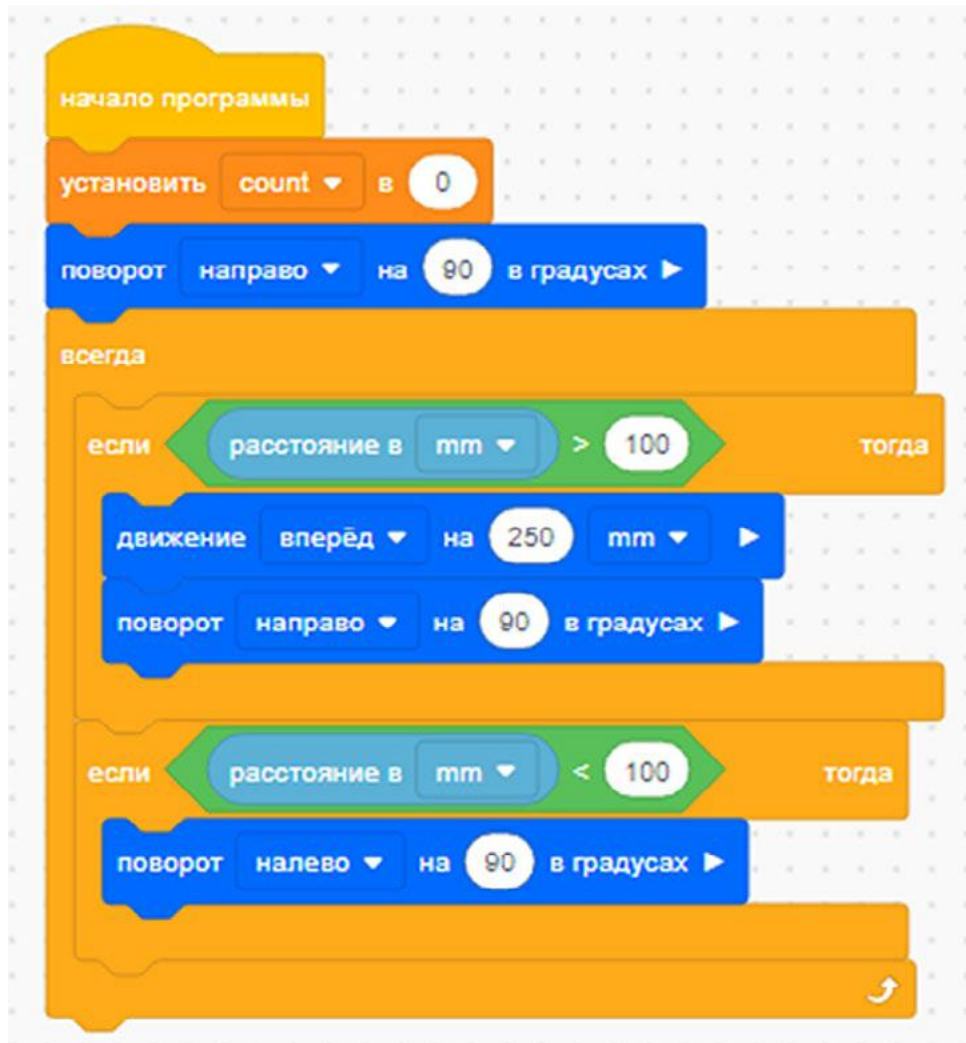


Рис. 35. Универсальная программа прохождения лабиринтов

Универсальность данной программы состоит в том, что VR-робот, перемещаясь на 1 клетку игрового поля, проверяет наличие препятствия на расстоянии 100 мм впереди себя и в зависимости от выполнения или невыполнения этого условия поворачивает в ту или иную сторону или движется вперед.

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 + 15 + 15 + 15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR, используя знания, полученные в ходе изложения нового материала. Данный этап урока занимает по 15 минут от каждого из четырех уроков.

В ходе отработки этого этапа урока учитель (в зависимости от подготовленности класса) предлагает учащимся совместно с ним выполнить практические задания.

Задание 1. Создать скрипт прохождения VR-роботом простого лабиринта (рис. 36) до каждой из букв на синем фоне.

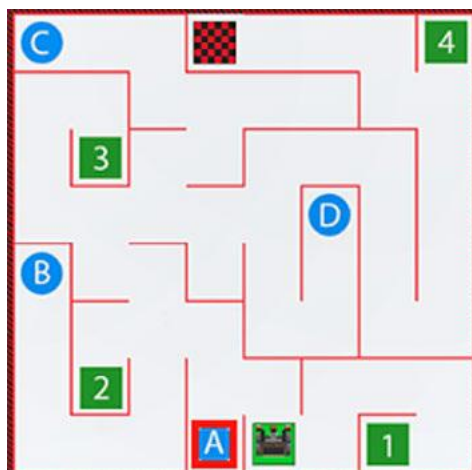
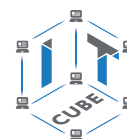


Рис. 36. Простой лабиринт

Задание 2. Оптимизировать скрипт, созданный в задании 1, по времени прохождения (фиксация таймером).

Важно!

Использовать различные датчики.

Задание 3. Создать скрипт прохождения VR-роботом простого лабиринта до каждой из цифр на зелёном фоне.

Задание 4*. Создать скрипты прохождения VR-роботом лабиринта с одновременным подсчётом количества синих и/или зелёных полей с выводом информации на консоль **Экрана**.

Деятельность учащихся: совместно с учителем выполняют предложенные задания.

Этап III. Повторение нового материала (5 + 5 + 5 + 5 мин)

Деятельность учителя: кратко повторяет новый материал: принцип работы датчиков цвета, касания, расстояния, местоположения, отражение данных на панели управления и консоли **Экрана**.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 + 5 + 5 + 5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Каков принцип работы датчика цвета?
- Каков принцип работы датчика расстояния?
- Как можно контролировать показания датчика?
- Каков принцип работы датчика касания?
- Каков принцип работы датчика местоположения?
- Как задать параметры датчика цвета?
- Как задать параметры местоположения?
- Как вывести данные на консоль **Экрана**?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 + 3 + 3 + 3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Написать программу движения VR-робота по дисковому лабиринту без использования блока **Если...то**.

Важно!

Использовать блок **Ждать пока**.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 + 2 + 2 + 2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXcode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Уроки 16–17

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 3. Датчики и обратная связь.

Тема урока: Управление магнитом. Сбор фишек.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с новой группой блоков управления роботом и возможностями программирования с их помощью.

Время реализации: 2 академических часа.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

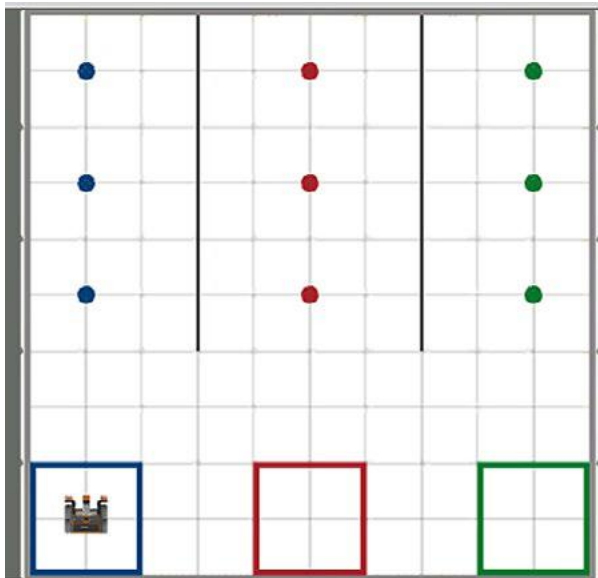


Рис. 37. Игровое поле **Перемещение фишек**

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности и изложение нового материала (15 + 15 + 15 + 15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией создания скрипта управления роботом на платформе VEXcode VR (<http://vr.vex.com>).

В группе **Магнит** всего один блок — **Активировать** или **Деактивировать** магнит. Давайте рассмотрим задание, в котором нужно за минимальное время собрать фишки в соответствующем цветовом поле (рис. 37). Для его выполнения нужно создать определённый скрипт.

Скрипт на перемещение фишек показан на рисунке 38.

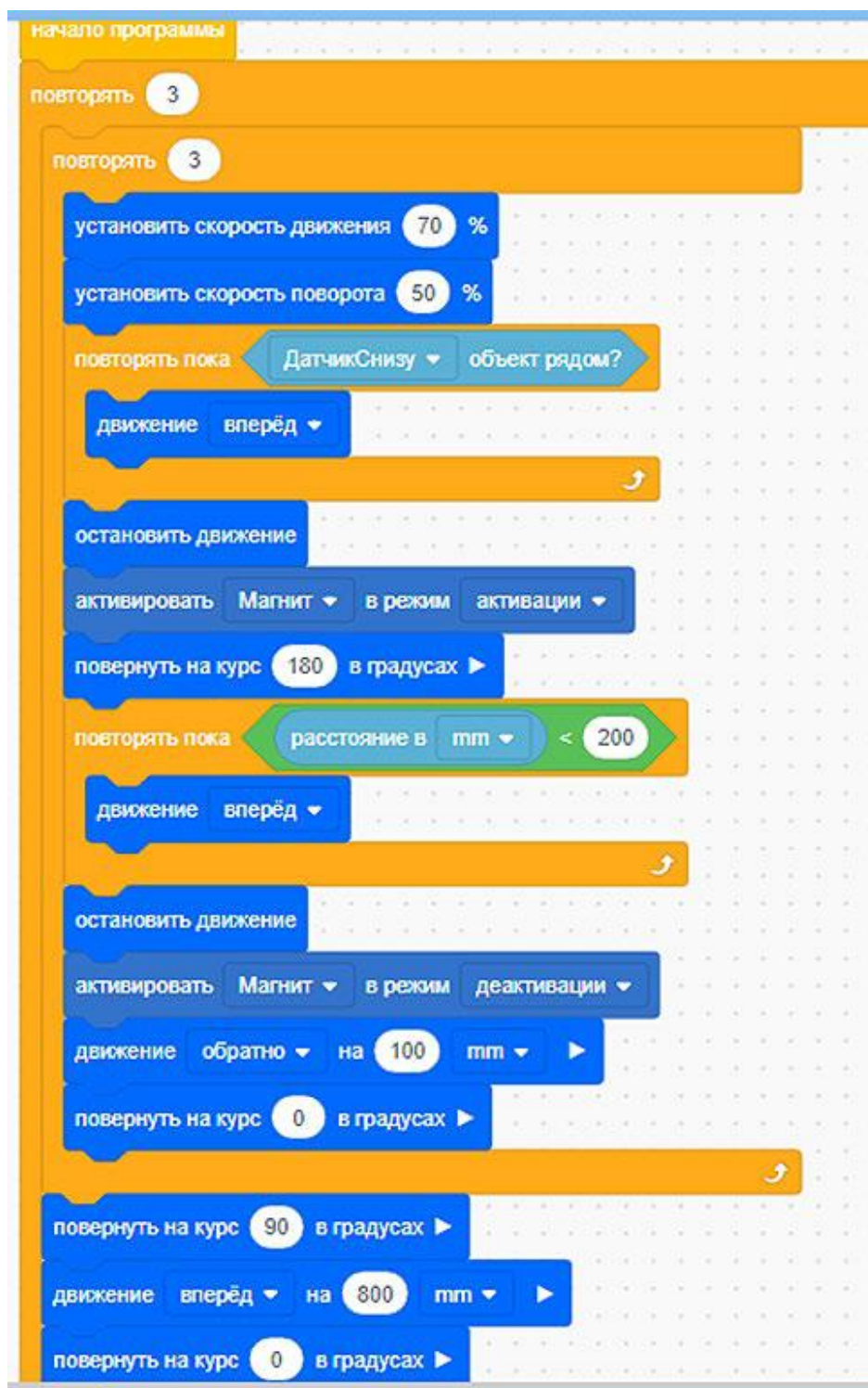
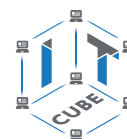


Рис. 38. Скрипт Перемещение фишек № 1

Учащимся необходимо пояснить данный алгоритм. В нём используется вложенный цикл повторения 3 раза (3 фишки и 3 поля). Для того чтобы обеспечить быстрое прохождение VR-робота данного игрового поля, сначала устанавливаем значение скорости движения и поворота в процентах от максимального значения. К сожалению, установить максимальное значение невозможно — робот проскочит фишку. Повороты VR-робота на 180° забирают лишнее время. Кроме того, в данном скрипте нет возможности остановить работу скрипта в конце последнего цикла.

В связи с этим рассматриваемый скрипт следует оптимизировать (рис. 39): VR-робот не должен терять время на развороты (задний ход), соответственно, можно увеличить его скорость, а также сделать фрагмент остановки робота после выполнения скрипта, используя переменную.

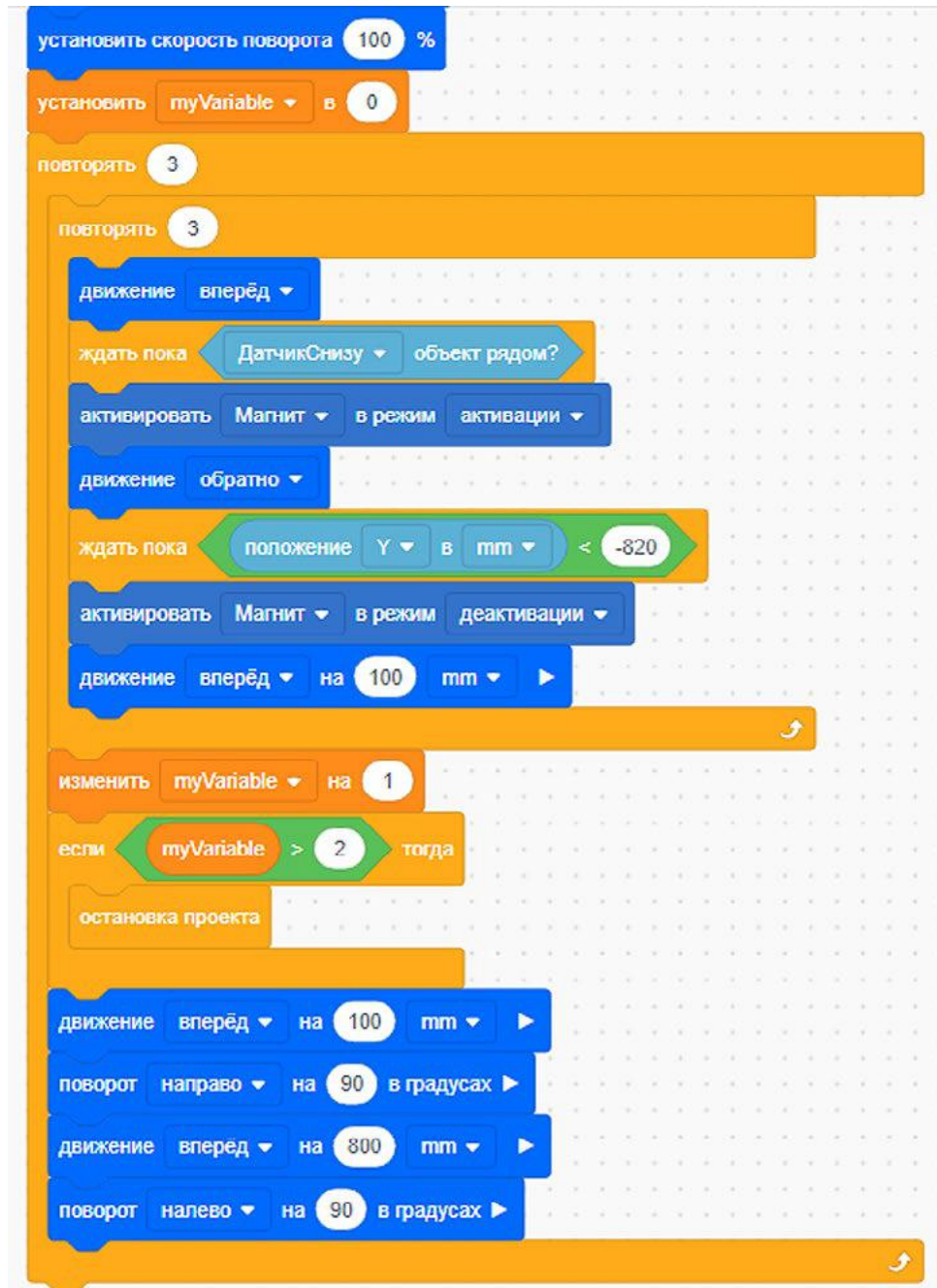
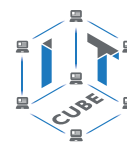


Рис. 39. Оптимизированный скрипт **Перемещение фишек № 2**

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 + 15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR, используя знания, полученные в ходе изложения нового материала.



Задание 1. Разобрать работу скрипта, приведённого на рисунке 36, выявить ошибки и неточности в его работе.

Задание 2. Разобрать работу скрипта, показанного на рисунке 37.

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Повторение нового материала (5 + 5 мин)

Деятельность учителя: кратко повторяет новый материал: принцип работы датчиков цвета, касания, расстояния, местоположения, отражение данных на панели управления и консоли **Экрана**.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичного закрепления (5 + 5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Каков принцип работы магнита?
- Как организовать вложенный цикл?
- Как регулировать скорость движения робота и скорость поворота?
- Как остановить робота после завершения работы скрипта?
- Как задать счётчик итераций (циклов) с помощью переменной?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 + 3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание*. Написать программу для игровой площадки **Транспортировка фишек**.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 + 2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXcode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 18

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 4. Реализация алгоритмов движения робота.

Тема урока: Блок команд **Управления** и организация циклов и ветвлений.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с ветвлениями и циклами на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией команд управления на платформе VEXcode VR (<http://vr.vex.com>).

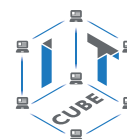
Блок команд Управления в VEXcode VR

Вы уже знакомы с понятиями циклов и ветвлений, поэтому самое время перейти к изучению этих структур в среде программирования VEXcode VR. Почему данная группа блоков имеет название **Управление**? На самом деле, всё довольно просто. Эти блоки позволяют управлять поведением нашего робота в зависимости от обстановки и окружающих его факторов. Без этих блоков мы не сможем получить интеллектуальную систему, способную ориентироваться в пространстве без дополнительной помощи со стороны оператора или команды разработчиков.



Рис. 40. Блок команд **Управления**

На рисунке 40 показан блок команд **Управления**. Обратите внимание, что многие блоки схожи между собой по названиям и обладают похожими свойствами. На уроке мы изучим блоки **Ждать** и **Повторять**.



Рассмотрим блок **Ожидания** (рис. 41).

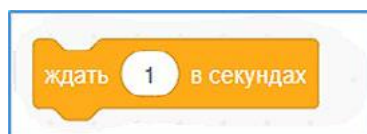


Рис. 41. Блок **Ожидания**

Данный блок позволяет удерживать действия, которые происходили до его вызова. Рассмотрим пример программы с использованием блока **Ожидания** (рис. 42).



Рис. 42. Пример программы с использованием блока **Ожидания**

Что произойдёт в данном отрывке кода? При запуске программы робот будет двигаться вперёд, после чего остановится. Если убрать данный блок, то визуально ничего не произойдёт, так как робот мгновенно выполнит обе команды и даже не сдвинется с места.

Важно!

Блоки программы после блока **Ожидания** не будут выполняться, пока не истечёт время, установленное внутри этого блока.

Перейдём теперь к блоку **Повторять**. Данный блок повторяет некоторые действия, количество которых определено внутри этого блока. Рассмотрим уже известную программу обхода поля по периметру квадрата (рис. 43).

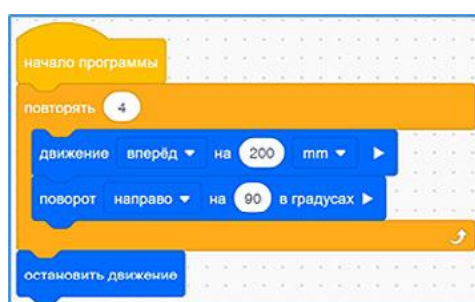


Рис. 43. Программа обхода поля по периметру квадрата

Давайте более подробно рассмотрим данную программу. Почему бы нам просто не использовать группу из одинаковых блоков для решения задачи? Никто не запрещает нам задействовать несколько одинаковых блоков вместо циклов. Однако при написании кода большего объёма получится гораздо больше строк, что, несомненно, усложнит читаемость всей программы. Для сравнения на рисунке 44 показан тот же самый код, но без использования цикла.

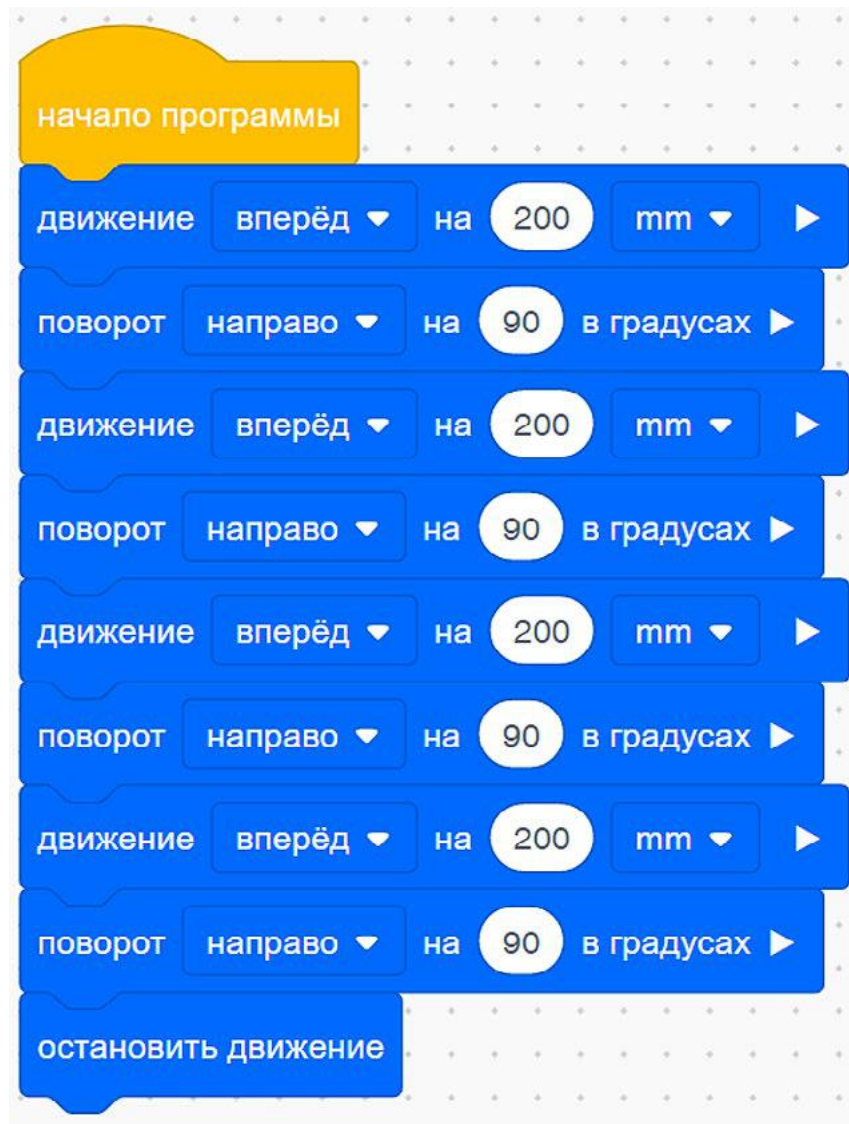


Рис. 44. Обход по квадрату без цикла

Как вы могли убедиться, циклы значительно упрощают читаемость кода, не влияя на его функциональность.

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание на платформе VEXcode VR, используя знания, полученные в ходе изложения нового материала.

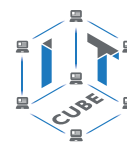
Задание. Написать программу для объезда VR-роботом шестиугольника, при этом используя только блоки движения без указания расстояния, блок поворота без указания угла и блок ожидания.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: блоки **Управления**, блок **Ожидания** и блок **Повторения**.

Деятельность учащихся: демонстрируют, как усвоили материал.

**Этап IV. Проверка понимания и первичного закрепления (5 мин)**

Деятельность учителя: задаёт учащимся следующие вопросы.

- Почему блоки управления так называются?
- Зачем нужны блоки **Управления**?
- Что позволяет делать блок **Ожидания**?
- Что делает блок **Повторения**?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание 1. Написать алгоритм объезда VR-роботом треугольника и прямоугольника.

Задание 2. Написать алгоритм объезда VR-роботом треугольника и прямоугольника без использования блоков поворота с указанием угла поворота.

Важно!

Необходимо использовать только блок ожидания.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXcode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 19

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 4. Реализация алгоритмов движения робота.

Тема урока: Блок команд **Управления** и организация циклов и ветвлений.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с ветвлениями и циклами на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока**Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)**

Деятельность учителя: объясняет новый материал с демонстрацией команд управления на платформе VEXcode VR (<http://vr.vex.com>).

Блок команд Управления в VEXcode VR

На предыдущем занятии были рассмотрены блоки ожидания и повторения. Сегодня мы продолжим изучение других блоков модуля управления. На практике роботу иногда требуется единожды принять выбор относительно будущего действия. Это могут быть повороты направо или налево, отъезды вперёд или назад в определённый момент времени без необходимости перепроверки действий, как в циклах. В этих случаях лучше всего использовать блоки **Если...тогда** и **Если...тогда...иначе**, которые позволяют преобразовать действия, совершаемые роботом, в систему с выходами «да/нет».

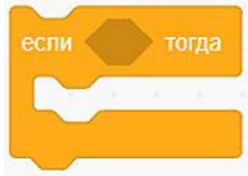


Рис. 45. Блок **Если...тогда**

В первую очередь, обратим внимание на блок **Если...тогда** (рис. 45).

Тело данного блока выполняется в том случае, если условие, обозначенное в шестиугольной вставке, является истиной, и не выполняется в обратном случае. Рассмотрим код, приведённый на рисунке 46.

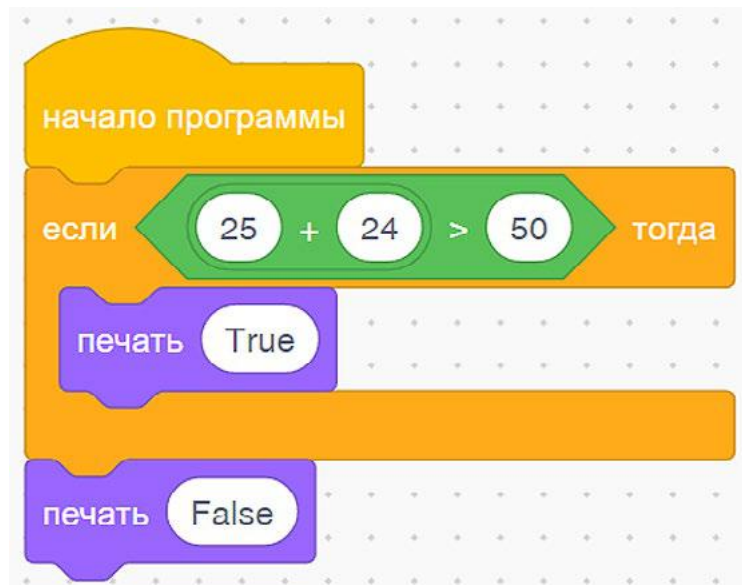


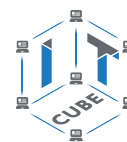
Рис. 46. Пример использования условного блока

В качестве условия в блоке дано логическое выражение, содержащее справа сумму чисел, а слева — константу. Если сравнение окажется истиной, то в окне вывода будет напечатано **True**, если условие окажется ложным, то **False**.

Кроме того, существует ещё один блок управления, схожий с вышеуказанным, но имеющий ещё одну дополнительную ветвь **Иначе** (рис. 47).



Рис. 47. Блок **Если...тогда...иначе**



Рассмотрим программу для понимания разницы между блоками **Если...тогда** и **Если...тогда...иначе** (рис. 48).

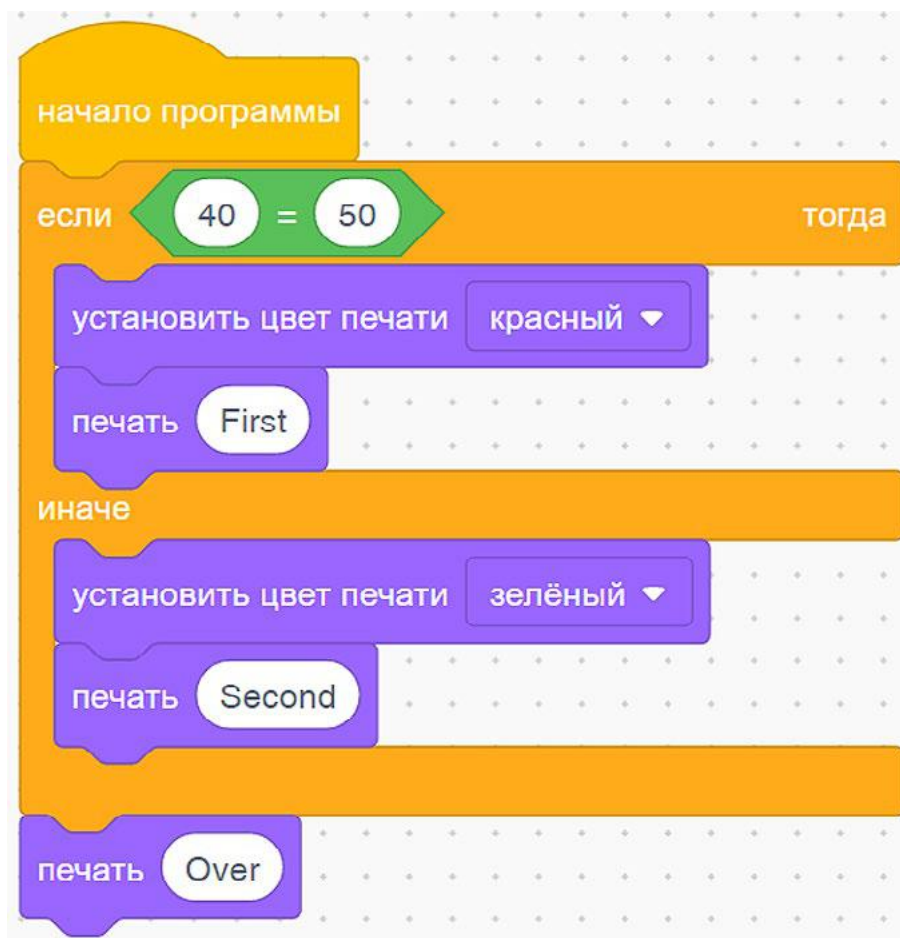


Рис. 48. Демонстрация работы программы **Если...тогда...иначе**

Рассмотрим код данной программы. Выражение после **Если** в данном случае является ложным, поэтому будет выполняться часть после **Иначе**, т. е. перо робота станет зелёного цвета и на консоли **Экрана** появится надпись **Second Over**. Обратите внимание, что **Over** будет выведено в любом случае, так как находится за пределами блока условного оператора.

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание на платформе VEXcode VR.

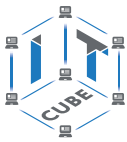
Задание. Написать программу для вывода на экран **True** или **False** в зависимости от решения простых математических выражений, которые учащиеся должны разместить внутри условия условного блока.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: блоки **Если...тогда**, **Если...тогда...иначе**.

Деятельность учащихся: демонстрируют, как усвоили материал.



Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задает учащимся следующие вопросы.

- Как можно использовать блок **Если...тогда**?
- В чём различие блоков **Если...тогда** и **Если...тогда...иначе**?
- В каких ситуациях лучше всего использовать блок **Если...тогда...иначе**?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание 1. Написать программу для остановки VR-робота перед препятствием.

Задание 2. Написать программу для поворота VR-робота перед препятствием на произвольный угол.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXcode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 20

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 4. Реализация алгоритмов движения робота.

Тема урока: Блок команд **Управления** и организация циклов и ветвлений.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с ветвлениями и циклами на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

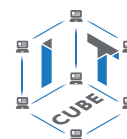
Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией команд управления на платформе VEXcode VR (<http://vr.vex.com>).

Блок команд Управления в VEXcode VR

На предыдущем занятии были рассмотрены условные блоки. Сегодня продолжим изучение остальных блоков модуля управления. На практике встречаются ситуации, при которых необходимо проверять условие несколько раз, прежде чем принять окончательное решение. Конечно, можно использовать совокупность блоков повторения и условные блоки,



но для корректной работы повторений должно быть огромное число (десятки миллионов), и подбор точного числа может значительно увеличить общее время создания программы. Во избежание этого в VEXcode VR используются циклы **Повторять пока** и **Пока** (рис. 49).

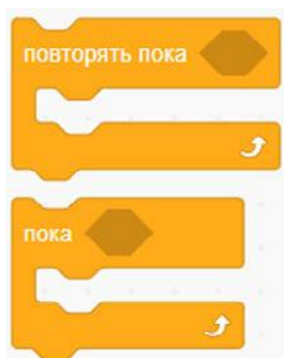


Рис. 49. Блоки циклов **Повторять пока** и **Пока**

Рассмотрим простой пример программы (рис. 50), демонстрирующий различия между этими двумя видами циклов.

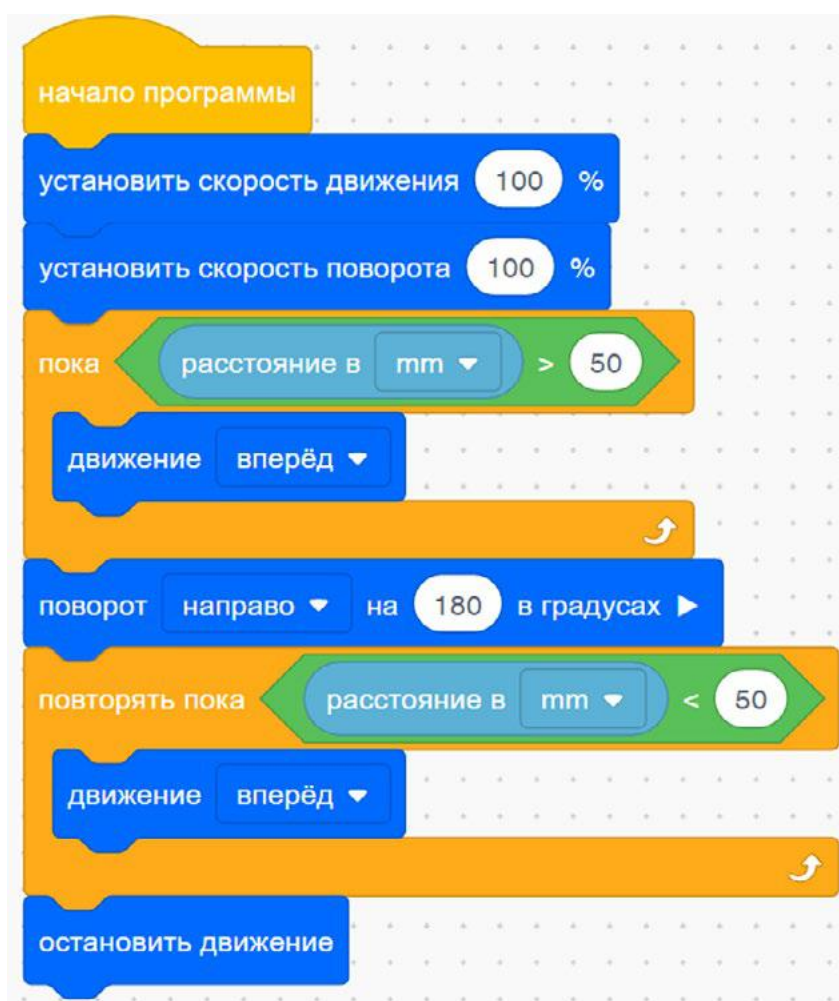


Рис. 50. Различия циклов **Повторять пока** и **Пока**

На первый взгляд может показаться, что различия в условиях блоков циклов должны привести к выполнению абсолютно разных действий, но это не так. Действия внутри бло-

ка **Повторять пока** будут выполняться до тех пор, пока условие будет ложным. Об этом важно помнить при составлении программы.

Давайте пока отложим блок **Повторять пока** и составим простую программу для вывода на экран нескольких строк текста в зависимости от показаний датчика расстояния (рис. 51).

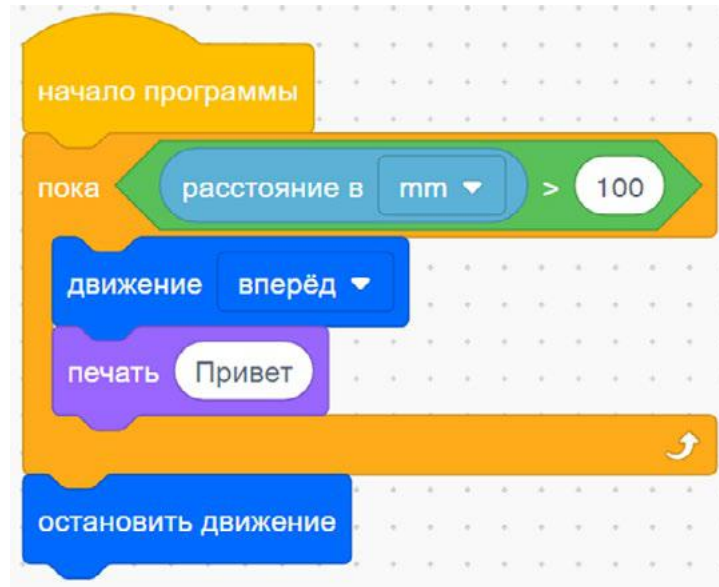


Рис. 51. Работа с циклом **Пока**

Построчно рассмотрим данную программу. В первую очередь создадим блок цикла, который будет работать до тех пор, пока робот не увидит перед собой какое-либо препятствие. На пути всего следования до препятствия робот будет выводить в консоль вывода слово «Привет» большое число раз в силу высокой производительности управляющего контроллера. Если добавить к программе всего одну строку, то слово «Привет» будет выведено только один раз (рис. 52).

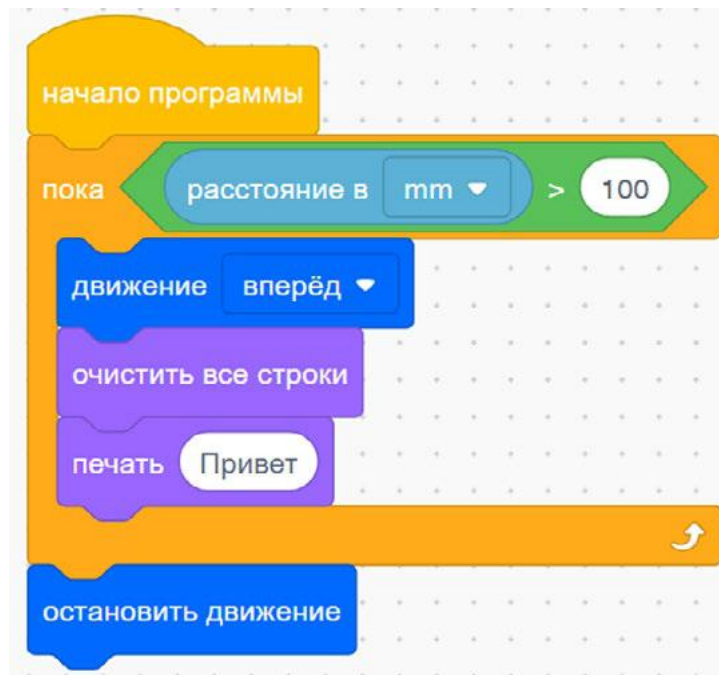
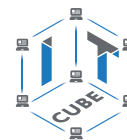


Рис. 52. Вывод одного слова на **Экран** во время использования цикла **Пока**



Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR.

Задание 1. Написать программу для вывода на экран слова «Привет».

Важно!

При каждом проходе цикла слово «Привет» должно выводиться с новой строки.

Задание 2. Написать программу по рисованию VR-роботом квадрата (в качестве повторения пройденного материала).

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: циклы в VEXcode VR.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- В чём различия блоков **Пока** и **Повторять пока**?
- Приведите примеры программы, использующей блок **Пока** (в виде плана работы программы).
- Приведите примеры программы, использующей блок **Повторять пока** (в виде псевдокода).

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Реализовать программу для рисования VR-роботом треугольника с выводом на экран всех этапов работы программы (повороты и движение вперёд).

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 21

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 4. Реализация алгоритмов движения робота.

Тема урока: Блок команд **Управления** и организация циклов и ветвлений.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с ветвлениями и циклами на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией команд Управления на платформе VEXcode VR (<http://vr.vex.com>).

Блок команд Управления в VEXcode VR

На предыдущем занятии были рассмотрены блоки циклов. Сегодня мы продолжим изучение оставшихся блоков модуля управления: блока **Всегда**, блока **Прерывания** и блока **Ждать пока**.

Блок **Всегда** (рис. 53) представляет собой бесконечный цикл без условия, который будет выполняться на протяжении всей программы независимо от действий, происходящих внутри и снаружи него.



Рис. 53. Блок цикла **Всегда**

Рассмотрим две программы, которые являются полностью идентичными, но при этом демонстрируют принцип работы блока **Всегда** (рис. 54).

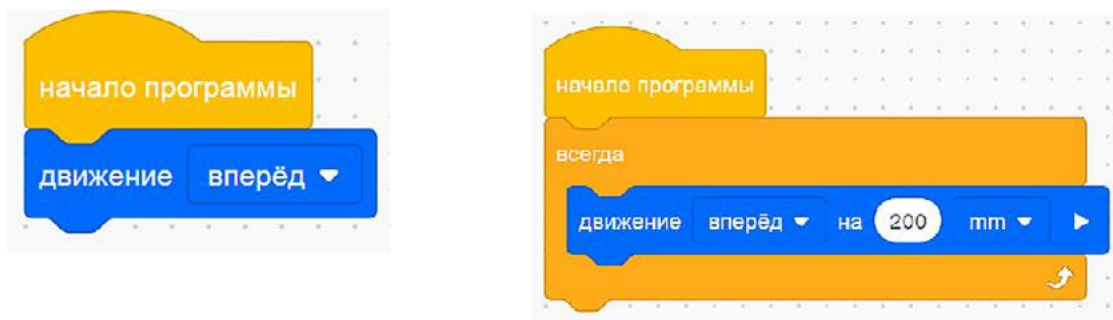


Рис. 54. Аналог бесконечного цикла

Как вы можете заметить, в данной программе блок движения включает в себя бесконечный цикл, что позволяет роботу двигаться бесконечно вперёд при добавлении этого блока в программу. Для того чтобы прервать выполнение бесконечного цикла, можно использовать блок **Прерывания** (рис. 55).

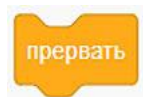


Рис. 55. Блок **Прерывания**

Рассмотрим программу, в которой робот будет ехать вперёд до тех пор, пока датчик расстояния не увидит перед собой некоторое препятствие на расстоянии 100 мм (рис. 56).

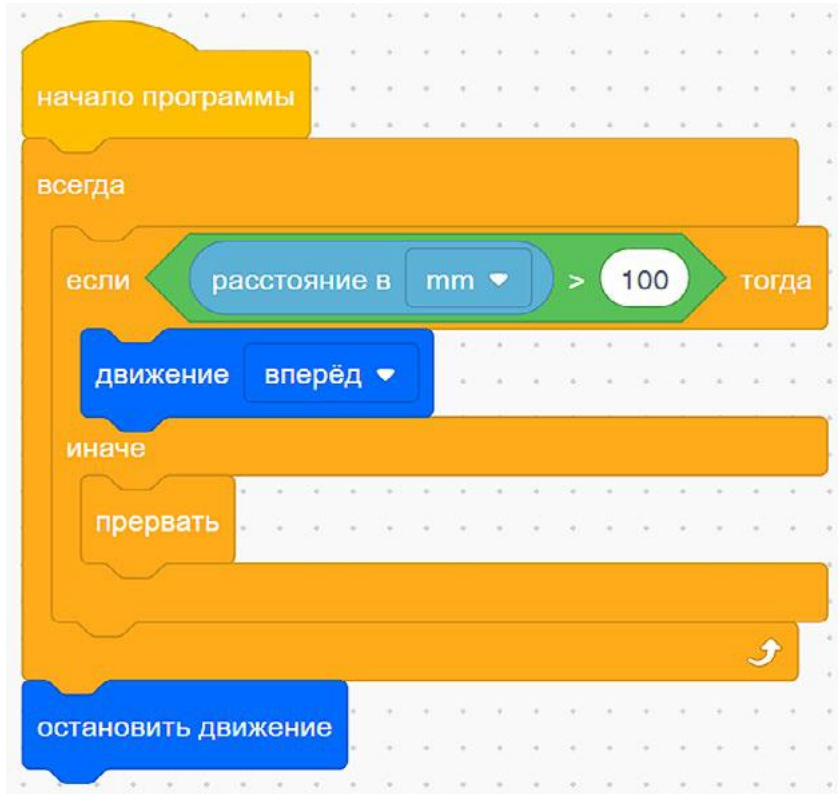


Рис. 56. Программа для демонстрации работы цикла прерывания

Данная программа выполняет все те же действия, что и использование цикла **Пока**, рассмотренного в предыдущем уроке, но при этом наглядно демонстрирует способ прерывания цикла **Всегда**. Блок **Прервать** обычно используется при аварийном прерывании программы, что позволяет избежать поломок робота.

Рассмотрим более сложный вариант программы, в котором использование цикла будет оправдано срабатыванием датчика расстояния на расстоянии меньшем, чем планируемое (рис. 57).

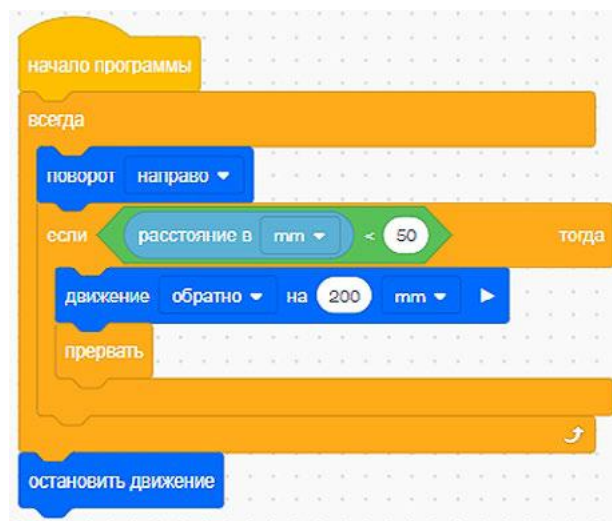


Рис. 57. Пример программы

В данной программе робот будет совершать поворот до тех пор, пока не обнаружит перед собой препятствие. После этого он остановит своё движение. Данный алгоритм может быть применим к случаям, когда на роботе установлено техническое зрение (камера) и ему необходим лучший обзор для идентификации объектов, находящихся перед ним.

Обратим внимание на блок **Ждать пока** (рис. 58), позволяющий удержать выполнение действий до активации этого блока.

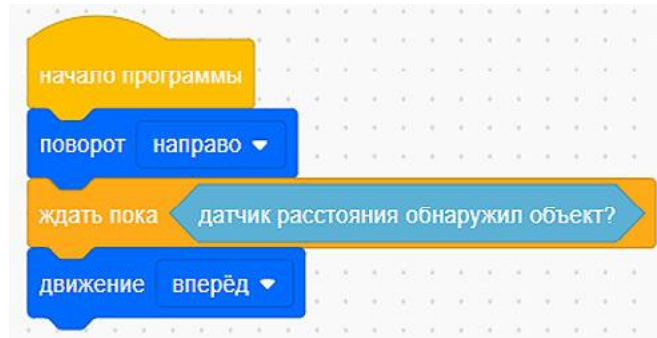


Рис. 58. Блок **Ждать пока**

Рассмотрим программу, которая облегчает прохождение роботом некоторых испытаний в среде VEXcode VR (рис. 59).

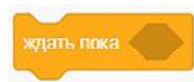


Рис. 59. Пример программы с блоком **Ждать пока**

В данной программе робот при обнаружении объекта постарается вытолкнуть его с поля. Данный код поможет в решении задач, с которыми вы встретитесь на следующих уроках.

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR.

Задание 1. Написать программу объезда VR-роботом всего периметра поля «Карта с разметкой» с использованием блока **Ждать пока**.

Задание 2. Написать программу по завершении VR-роботом действий при обнаружении под собой красного цвета.

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: блоки **Прервать**, **Всегда** и **Ждать пока**.

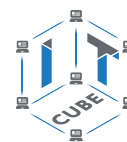
Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Зачем нужен блок **Прервать**?
- Приведите примеры программы, использующей блок **Прервать** (в виде псевдокода).
- Как работает блок **Ждать пока**?

Деятельность учащихся: отвечают на предложенные вопросы.



Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Реализовать программу для активации VR-роботом отъезда назад при его первоначальном движении вперёд.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXcode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 22

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 4. Реализация алгоритмов движения робота.

Тема урока: Проекты «Разрушение замка» и «Динамическое разрушение замка».

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с выполнением проектов по уборке территории на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией команд управления на платформе VEXcode VR (<http://vr.vex.com>).

Создание собственных блоков

На практике часто встречаются ситуации, когда некоторые блоки программы повторяются несколько раз во время написания кода. Поэтому разумнее всего сформировать из них отдельные блоки, которые позволят улучшить читабельность кода и сократить его объём. В VEXcode VR можно создавать собственные блоки, обладающие уникальными свойствами и наборами действий. Для этого необходимо на левой панели нажать на кнопку **Создать блок**. В результате появится окно (рис. 60), в нём необходимо указать количество данных, которые будут начальными при обращении к этому блоку. Это могут быть числовые данные, логические данные (**True** или **False**) или же просто пояснения к этому блоку.

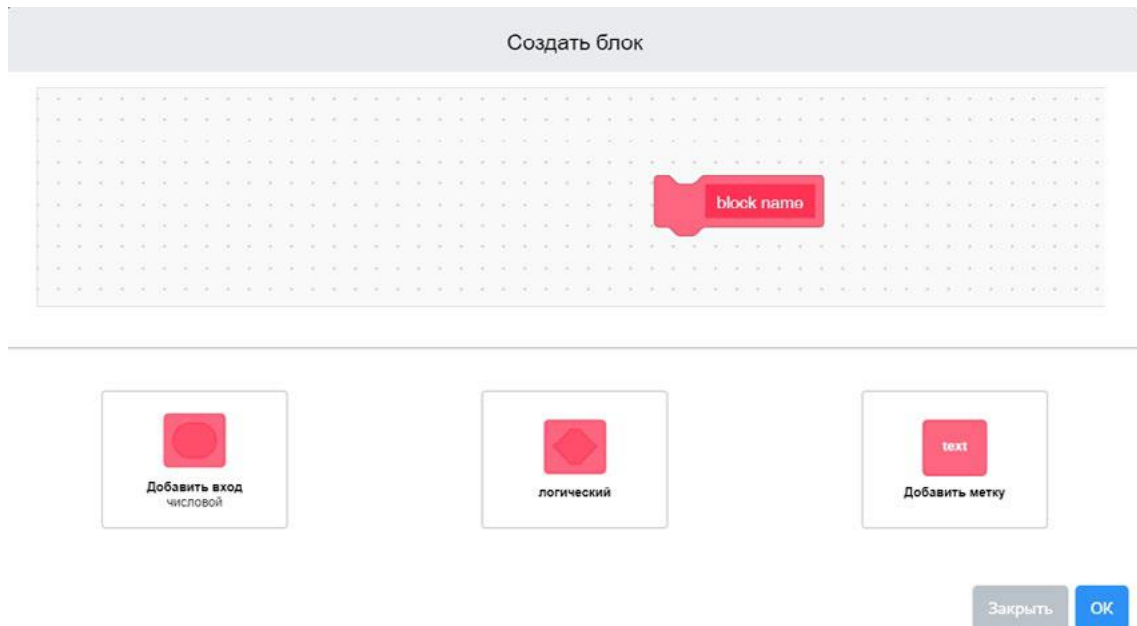


Рис. 60. Процесс создания собственного блока

Каким образом будет использован инструмент для создания собственного блока при написании программы для проекта «Разрушение замка»? Самым простым способом по очистке территории от замка является последовательный проезд VR-робота по всем ключевым точкам, на которых установлены элементы замка.

Алгоритм по очистке блоков замка может быть самым разным в зависимости от выбора стратегии. Например, VR-робот может последовательно объехать весь периметр (рис. 61).

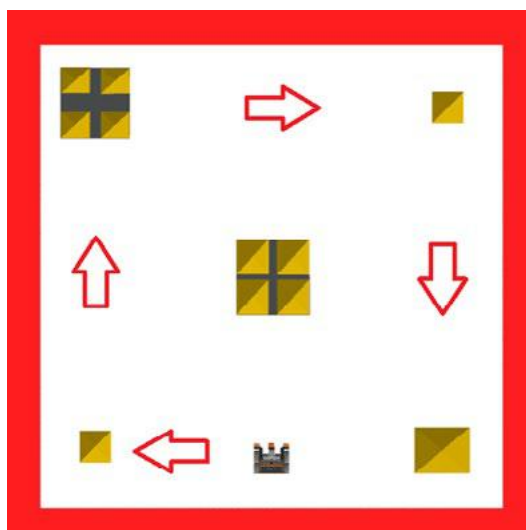


Рис. 61. Стратегия объезда VR-роботом препятствий

Как же лучше всего использовать инструмент для создания собственных блоков при такой стратегии? При составлении программы необходимо на вход блока подать данные о расстоянии, на которое необходимо будет проехать VR-роботу до блока, и угол поворота, на который ему будет необходимо повернуться.

При следовании стратегии, изображённой на рисунке 59, VR-роботу необходимо поворачивать только на 90 градусов направо, поэтому новый блок примет вид, показанный на рисунке 62.

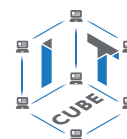


Рис. 62. Блок для перемещения VR-робота по полю

Использование этого блока несущественно упростит нашу программу на первом этапе, но при этом облегчит её дальнейшее написание. Давайте составим программу для удаления с поля нижнего правого блока замка на поле (рис. 63).



Рис. 63. Программа по удалению нижнего правого блока

Данная программа начинается с небольшого перемещения VR-робота вперёд, так как для удаления блока необходимо выравнивание робота относительно этого блока, и далее уже идёт само удаление блока, т. е. сброс за пределы поля.

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание на платформе VEXcode VR.

Задание. Написать программу по последовательному удалению с поля правых нижнего и верхнего блоков.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: инструмент для создания собственных блоков.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Как создать собственный блок?
- Какую стратегию объезда лучше всего выбрать?
- Каким образом добавить на вход собственного блока новые значения?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание 1. Добавить на вход собственного блока угол поворота VR-робота.

Задание 2. Исправить программу с занятия в соответствии с входными данными для нового блока.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXcode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 23

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 4. Реализация алгоритмов движения робота.

Тема урока: Проекты «Разрушение замка» и «Динамическое разрушение замка».

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с выполнением проектов по уборке территории на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока**Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)**

Деятельность учителя: объясняет новый материал с демонстрацией команд управления на платформе VEXcode VR (<http://vr.vex.com>).

Рассмотрение дополнительных вариантов стратегий

На предыдущем уроке была рассмотрена всего одна стратегия по уборке территории от обломков замка. На сегодняшнем занятии будут изучены дополнительные блоки и расширена функциональность нашего блока для успешного выполнения всех заданий.

В первую очередь давайте рассмотрим блок **Повернуть на курс** (рис. 64), который является компасом, позволяющим VR-роботу в любой момент времени поворачивать в определённую сторону.



Рис. 64. Блок Повернуть на курс

Рассмотрим пример использования этого блока по аналогии с другими блоками поворотов в VEXcode VR. Давайте вспомним градусную меру в рамках так называемого тригонометрического круга (рис. 65). Обратите внимание, что на платформе VEXcode VR вращение по часовой стрелке принимается за положительное, а вращение против часовой стрелки — за отрицательное (в отличие от классического представления тригонометрического круга).

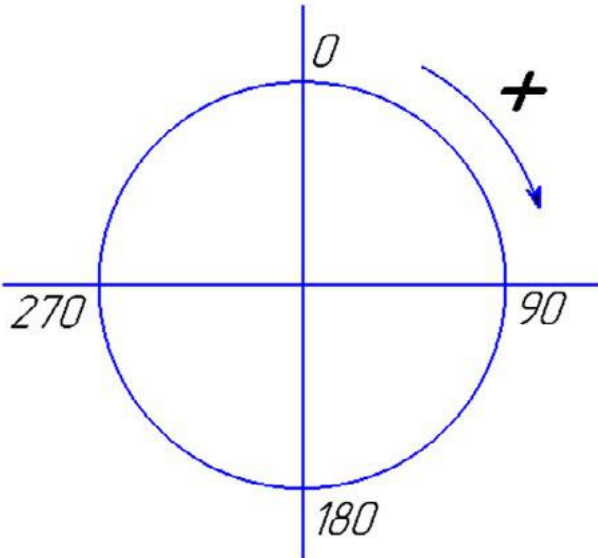


Рис. 65. Тригонометрический круг поворота робота

В чём же заключается смысл работы блока **Поворот на курс**? Для ответа на этот вопрос рассмотрим программу, показанную на рисунке 66.

```

начало программы
поворот направо на 147 в градусах
повернуть на курс 90 в градусах
поворот направо
ждать 1 в секундах
повернуть на курс 90 в градусах
    
```

Рис. 66. Пример программы с использованием блока **Поворот на курс**

Для этой программы были выбраны случайные значения поворота VR-робота, так как они наглядно демонстрируют принцип работы блока **Поворот на курс**. После каждого выполнения поворота на случайный угол VR-робот всегда будет возвращаться к углу в 90 градусов (или осуществлять поворот на восток). Данный блок позволяет осуществлять более точную навигацию по полю и помогает VR-роботу всегда приходить в нужное положение в случае потери ориентира.

Целесообразно модифицировать наш собственный блок так, чтобы он мог обрабатывать команду **Поворот на курс** (рис. 67).

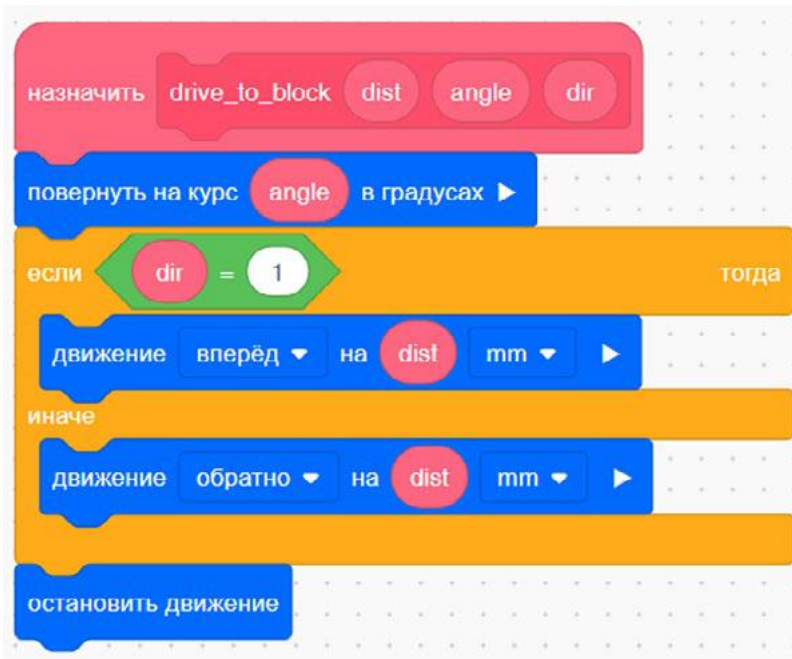


Рис. 67. Модифицированный блок движения

Для удобства на вход подаётся угол поворота VR-робота курс для того, чтобы разнообразить его движение и увеличить вероятность успеха очистки территории. Кроме того, необходимо ввести числовую переменную **dir**, которая позволит сообщать блоку о требуемом направлении движения (вперёд или назад). Если значение переменной **dir** будет равно 1, то VR-робот поедет вперёд, если любому другому числу, то — назад.

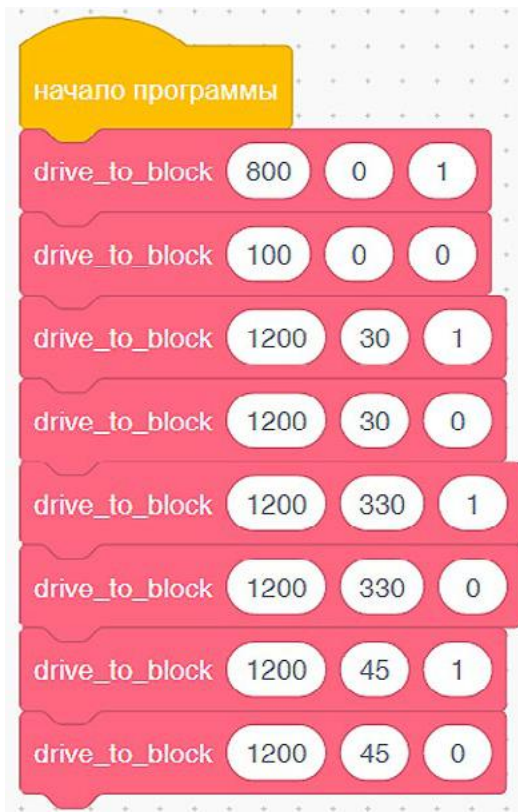
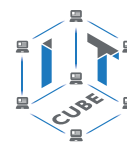


Рис. 68. Реализация алгоритма по очистке территории

В качестве новой стратегии предлагается начать очистку территории от обломков замка с территории центрального блока. Основная сложность заключается в том, что центральный блок состоит из множества блоков, и в случае неудачи эти блоки могут распасться и усложнить задачу. Во избежание этого необходимо грамотно подобрать параметр скорости движения VR-робота и доработать алгоритм (рис. 68) для расталкивания блоков от центра к краям в стороны. Ключевым моментом является отслеживание правильных углов поворота и расстояний, на которые будет проезжать VR-робот.

Что же происходит в рамках данной программы? VR-робот доезжает до центра и начинает понемногу поворачивать на определённый угол для очистки всего периметра от обломков.

На следующий занятиях алгоритм будет модифицирован таким образом, чтобы VR-робот поворачивал на определённый угол с помощью понятия «шаг цикла».



Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание на платформе VEXcode VR.

Задание. Написать программу по последовательному удалению центрального и верхних блоков.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: дополнительные параметры блоков, блок поворота на курс.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- В чём отличие блока **Поворот на курс** от других блоков?
- Какую стратегию объезда лучше всего выбрать?
- Зачем к собственному блоку необходимо добавить дополнительные входы?
- Какие ещё входы можно добавить к нашему блоку?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Написать программу по очистке нижней линии блоков.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 24

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 4. Реализация алгоритмов движения робота.

Тема урока: Проекты «Разрушение замка» и «Динамическое разрушение замка».

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с выполнением проектов по уборке территории на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией команд управления на платформе VEXcode VR (<http://vr.vex.com>).

Использование переменных для улучшения алгоритма по очистке территории

На предыдущем занятии был рассмотрен алгоритм по очистке территории с использованием собственных блоков. При этом вы могли заметить, что изменение вручную углов и расстояний является довольно трудоёмким процессом, который занимает много времени. Кроме того, при необходимости изменения значений в блоках приходится менять значения построчно, что также требует дополнительного времени при разработке программы. Для того чтобы этого избежать, необходимо использовать блок **Создание переменных**.

Для создания новой переменной следует нажать на соответствующую кнопку в правом меню — **Создать переменную**. Перед вами откроется окно (рис. 69), в котором необходимо задать имя переменной.

Давайте создадим переменную, в которой будем хранить необходимое расстояние проезда VR-робота вперёд при отклонении от курса. Это вполне логичный шаг,

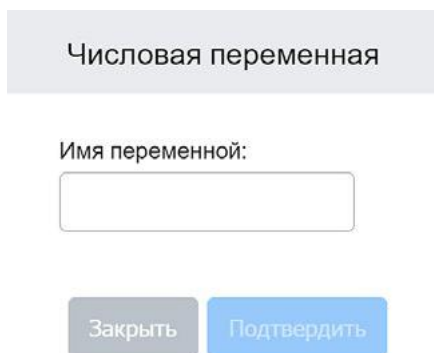
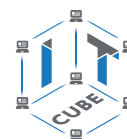


Рис. 69. Окно создания переменной



Рис. 70. Общий вид программы после ввода переменной



так как значение расстояния не меняется в ходе выполнения программы и в случае необходимости его можно будет изменить всего в одном месте программы при так называемой инициализации переменной.

После ввода переменной программа примет вид, показанный на рисунке 70.

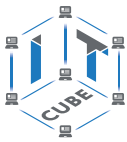
Что ещё можно улучшить в разработанной нами программе? Прежде всего напрашивается пошаговое изменение градусной меры без необходимости ручного ввода значений угла во время каждого поворота. Для реализации этой идеи нужно ввести две переменные: шаг и угол поворота. Шаг — это некоторое значение, на которое будет изменён угол поворота VR-робота. Для изменения шага необходимо применить цикл с повторениями, который будет назначен единожды в условии цикла **Повторения**. Итоговый вид программы изображён на рисунке 71.



Рис. 71. Программа с шагом угла

Рассмотрим эту программу более подробно. В первую очередь, необходимо создать переменные: шага, угла и расстояния. После этого VR-робот должен проехать до центра поля и растолкать центральный блок с небольшим отъездом назад для центровки. В цикле будет совершён последовательный проезд относительно оси симметрии поля вправо и влево, изменено значение угла.

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.



Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание на платформе VEXcode VR.

Задание. Написать программу по последовательному удалению верхних блоков.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: создание переменных, использование переменных.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Каким образом создаётся новая переменная?
- Что такое шаг цикла и зачем он нужен?
- Приведите примеры собственных алгоритмов с использованием шага.

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Написать программу по очистке правой линии блоков.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 25

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения: Модуль 4. Реализация алгоритмов движения робота.

Тема урока: Проекты «Разрушение замка» и «Динамическое разрушение замка».

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с выполнением проектов по уборке территории на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией команд управления на платформе VEXcode VR (<http://vr.vex.com>).

Использование датчиков для улучшения алгоритма по очистке территории

На предыдущем занятии был рассмотрен алгоритм по очистке поля от обломков замка с использованием только блоков поворота, движения и понятия шага. Теперь настало время предоставить VR-роботу возможность самостоятельно принимать решения, основываясь на показаниях датчиков.

В первую очередь следует рассмотреть применение датчика расстояния для определения VR-роботом наличия объектов перед ним. Алгоритм выполнения такой программы достаточно прост: пока VR-робот будет видеть перед собой объект, он будет ехать вперёд. Для того чтобы более целесообразно использовать данный алгоритм, необходимо внести изменения в блок движения (рис. 72).

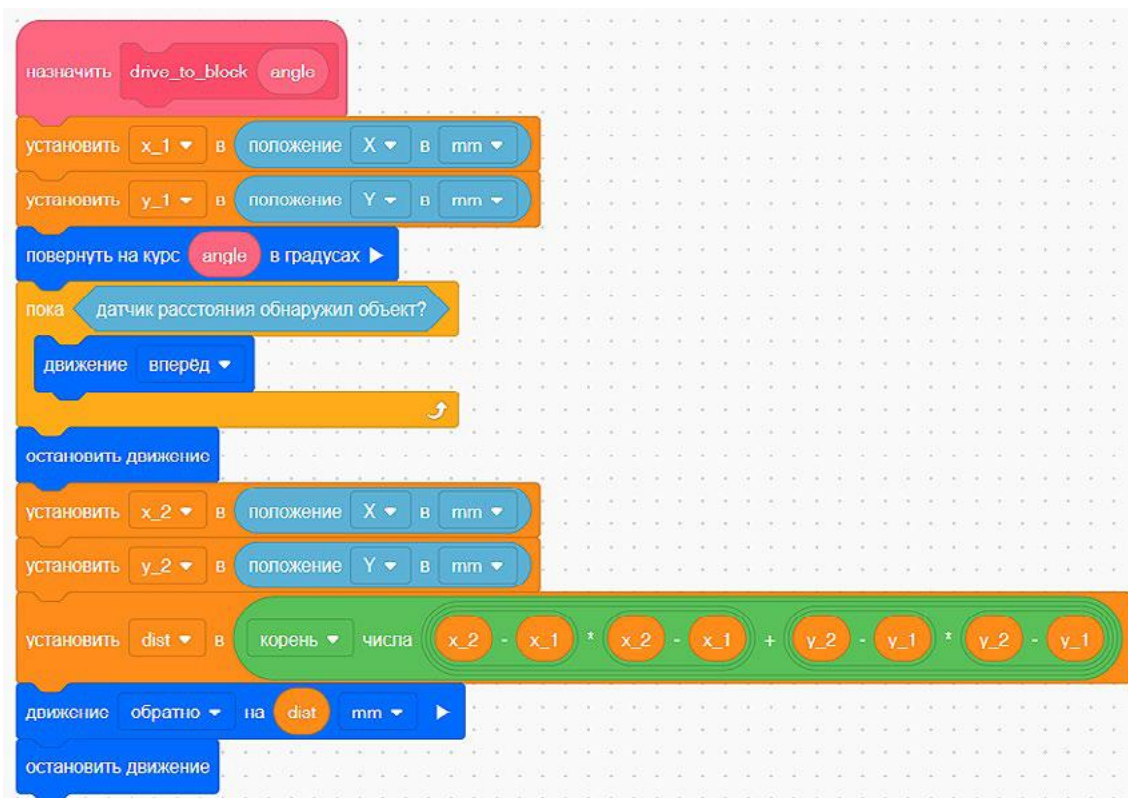


Рис. 72. Модификация блока движения

Целесообразно более детально описать этот блок, чтобы все его компоненты стали предельно ясны. Прежде всего нас интересуют блоки **Координат** (рис. 73), которые позволяют определять текущее положение VR-робота на поле.

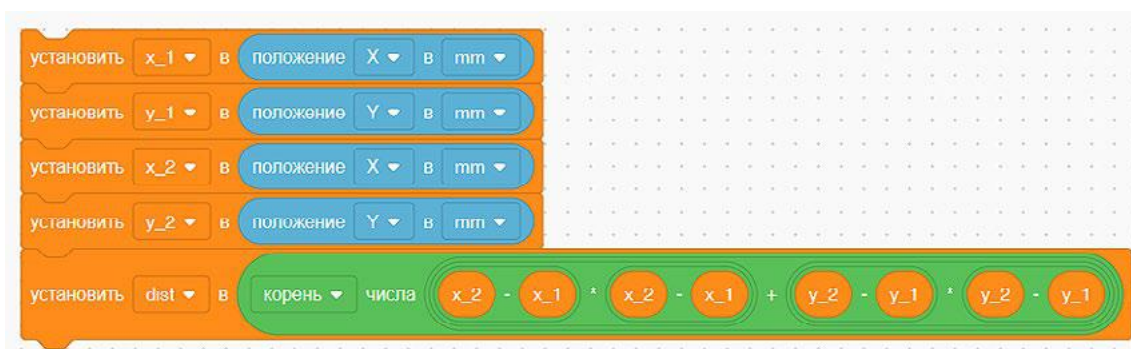


Рис. 73. Использование блоков Координат

Переменные x и y с индексами 1 и 2 характеризуют текущее и конечное положения VR-робота на поле. Знание этих координат позволяет вычислить конечное расстояние, на которое VR-роботу необходимо будет вернуться для повторения алгоритма по очистке поля.

Интерес также представляет цикл **Пока** (рис. 74), указывающий на необходимость движения VR-робота до тех пор, пока перед ним не закончатся объекты.

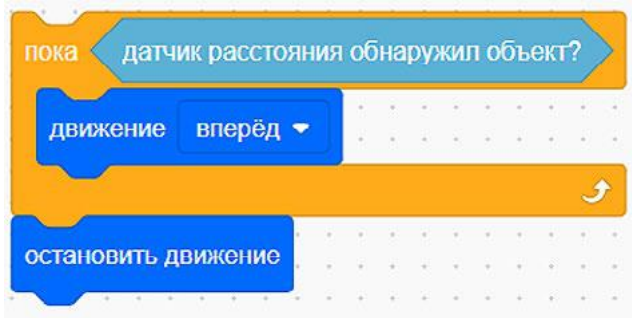
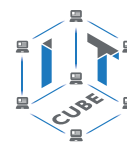


Рис. 74. Цикл **Пока** для движения VR-робота при наличии объекта

Таким образом, мы получили универсальный блок для осуществления VR-роботом уборки территории от обломков замка. Пример итоговой программы для прохождения VR-роботом этого испытания показан на рисунке 75.



Рис. 75. Пример итоговой программы для прохождения VR-роботом испытания



Обратите внимание на необходимость вычитания угла из 360 градусов для автоматического поворота VR-робота налево.

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание на платформе VEXcode VR.

Задание. Написать программу по последовательному удалению всех блоков поля.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: блоки для работы с датчиками.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: предлагает учащимся выполнить следующие задания.

- Предложите модификации для собственного блока.
- Предложите свою стратегию по уборке поля от обломков замка.
- Приведите примеры собственных алгоритмов с использованием шага.

Деятельность учащихся: выполняют предложенные задания.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Написать программу для уборки центрального блока.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 26

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 4. Реализация алгоритмов движения робота.

Тема урока: Проект «Детектор линии».

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с выполнением проектов по обнаружению линий на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией команд управления на платформе VEXcode VR (<http://vr.vex.com>).

Подсчёт количества линий

Одной из самых распространённых задач в соревновательной робототехнике является задача по обнаружению границ чёрного и белого, что позволяет роботу корректно ездить по линии. После решения этой задачи обычно переходят к задачам по распознаванию штрих-кодов.

Что же из себя представляет обычный штрих-код (рис. 76)?

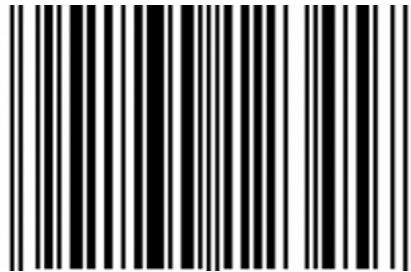


Рис. 76. Пример штрих-кода

Фактически это всего лишь набор линий, расположенных на некотором расстоянии друг от друга. Меняя расстояние между линиями, можно закодировать сообщения о названиях товаров, городов или услуг. На сегодняшнем уроке будет рассмотрена задача по обнаружению и подсчёту количества линий на карте **Детектор линий** (рис. 77).

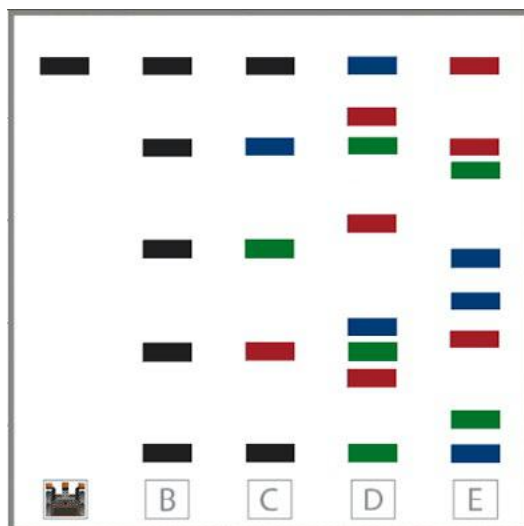


Рис. 77. Карта Детектор линий

Попробуем считать первую линию на дорожке А и выведем на экран соответствующее сообщение (рис. 78).

В данной программе VR-робот будет ехать вперёд до тех пор, пока не обнаружит на своём пути чёрную линию, которая характеризуется параметром «не никакой» в блоке **Обнаружения цвета**.

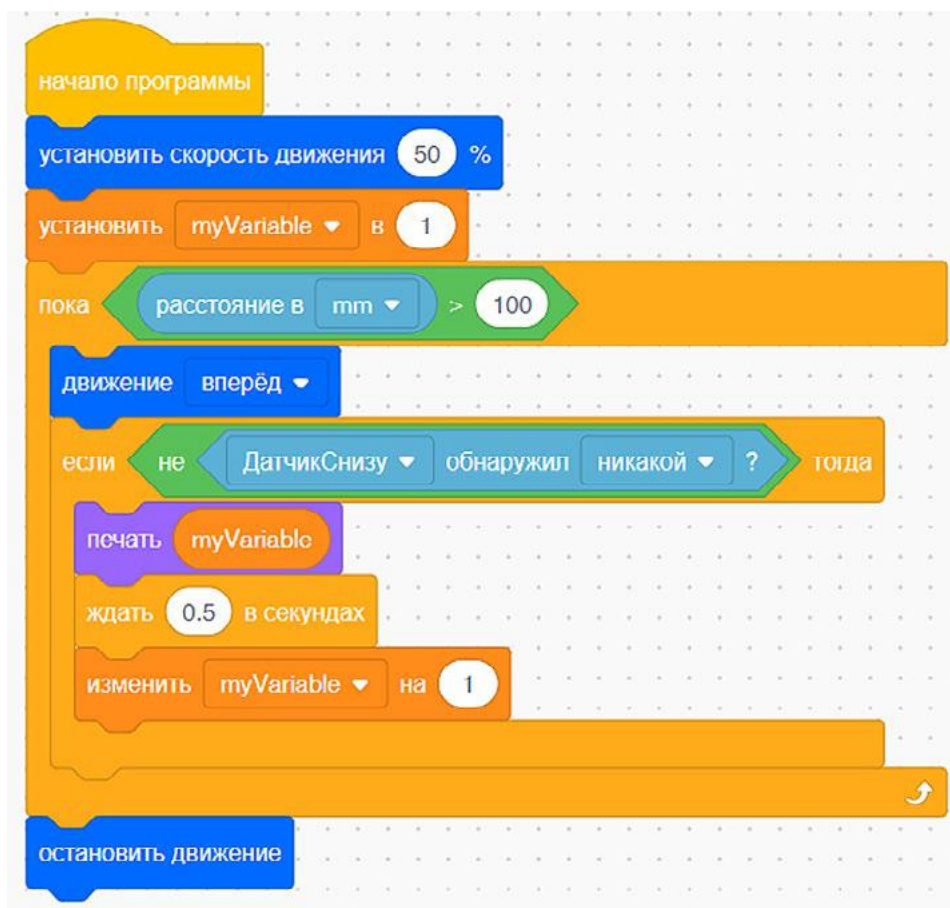
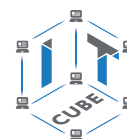


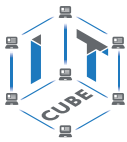
Рис. 78. Программа для поиска линии

Давайте теперь усовершенствуем эту программу. При каждом обнаружении линии любого цвета VR-робот будет выводить на экран номер этой линии (рис. 79).



Рис. 79. Программа для подсчёта количества линий

В данной программе введена отдельная переменная **myVariable** для того, чтобы вести подсчёт линий, появляющихся под датчиком. Внешний цикл **Пока** необходим для того, чтобы VR-робот остановился у дальней стены после пересчёта линий. Блок условий **Если...тогда** позволяет VR-роботу при наличии линии любого цвета, отличного от белого, увеличивать значение переменной и выводить его в **консоль**. Задержка в виде блока **Ожидания** позволяет избегать повторного вывода показаний датчика.



Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание на платформе VEXcode VR.

Задание. Написать программу по подсчёту только красных линий.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: блоки для взаимодействия с датчиками, алгоритм подсчёта линий.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Каким образом можно кодировать сообщения с помощью линий?
- Какую переменную необходимо дополнительно ввести в программу, чтобы улучшить её?
- Как выглядит алгоритм подсчёта чёрных линий?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Написать программу по подсчёту только чёрных линий.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 27

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 4. Реализация алгоритмов движения робота.

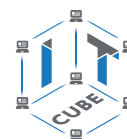
Тема урока: Проект «Детектор линии».

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с выполнением проектов по обнаружению линий на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.



Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: объясняет новый материал с демонстрацией команд управления на платформе VEXcode VR (<http://vr.vex.com>).

Подсчёт линий разного цвета

На предыдущем занятии были рассмотрены алгоритм и примеры программ для обнаружения и подсчёта линий на поле «Детектор линий». Одним из ключевых моментов для изучения этого материала является широкое применение так называемых штрих-кодов, которые могут хранить информацию об изделиях, местоположениях и многом другом.

На сегодняшнем уроке мы рассмотрим подход к подсчёту разных цветов и преобразования этой информации в вывод. В качестве вывода в консоль будет выводиться буква дорожки, на которой в данный момент находится VR-робот. Прежде всего необходимо создать собственный блок, который будет принимать на вход количества каждого из цветов и в зависимости от этого делать вывод на экран (рис. 80).

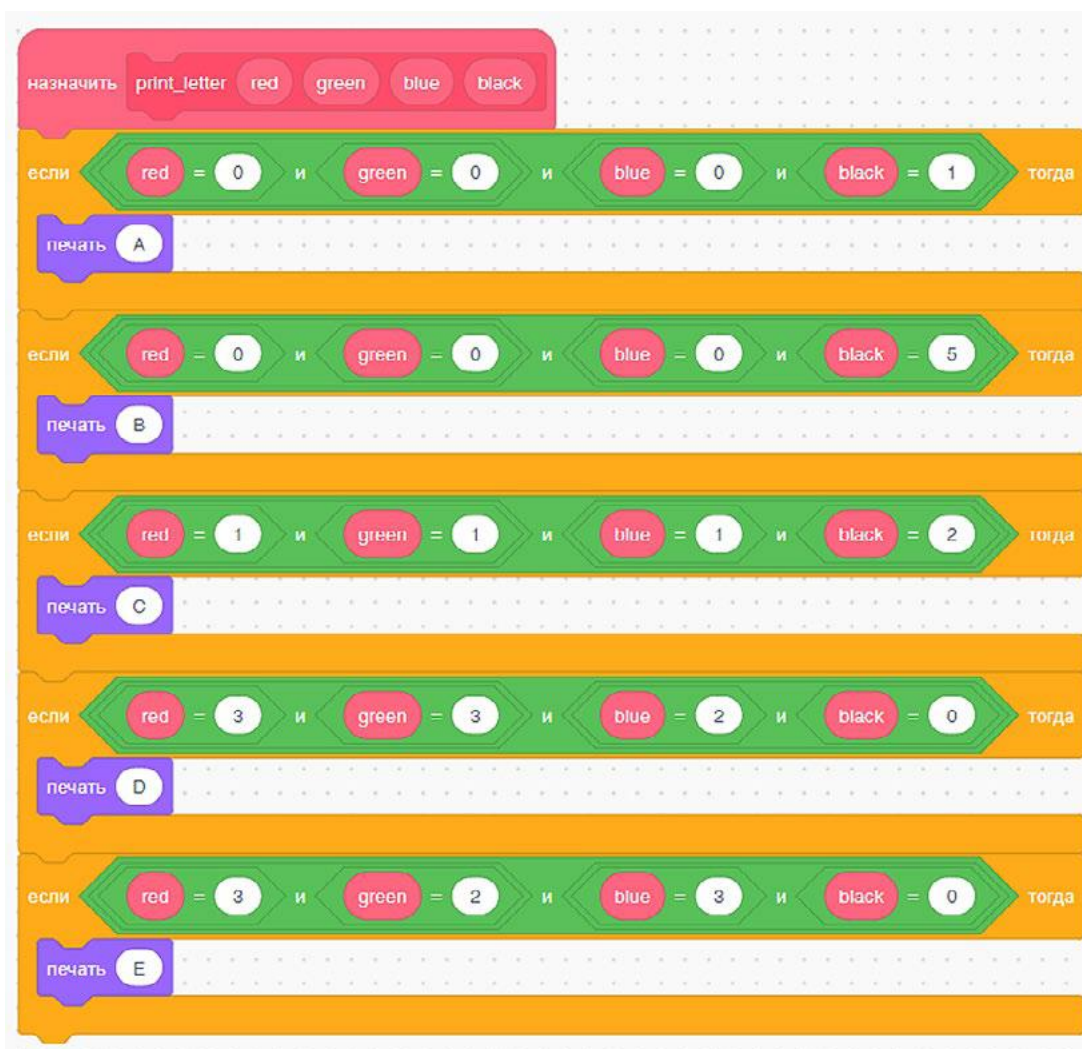


Рис. 80. Блок вывода букв

Как можно заметить, данный блок просто подводит итог выполнения программы, используя блок **Если...тогда** для принятия решений о выводе. Помимо этого блока, нам также понадобится блок, который будет собирать информацию о количестве линий разных цветов (рис. 81).

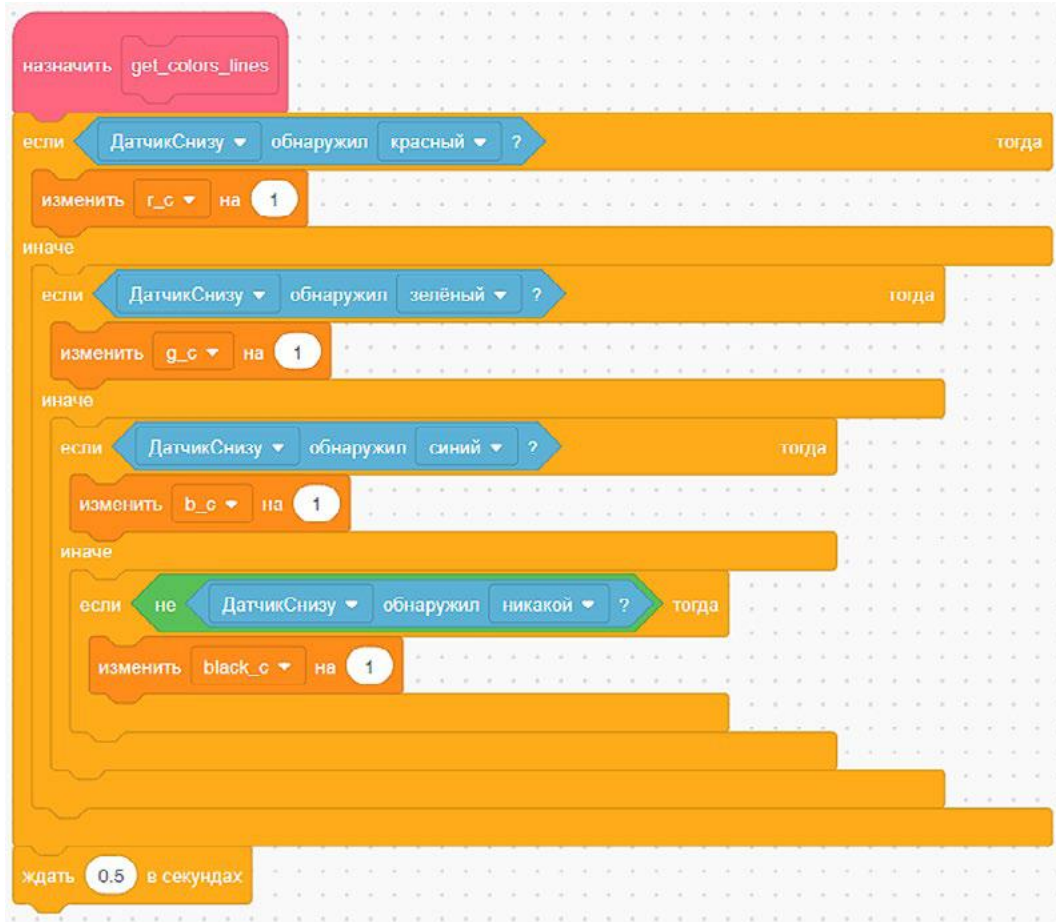
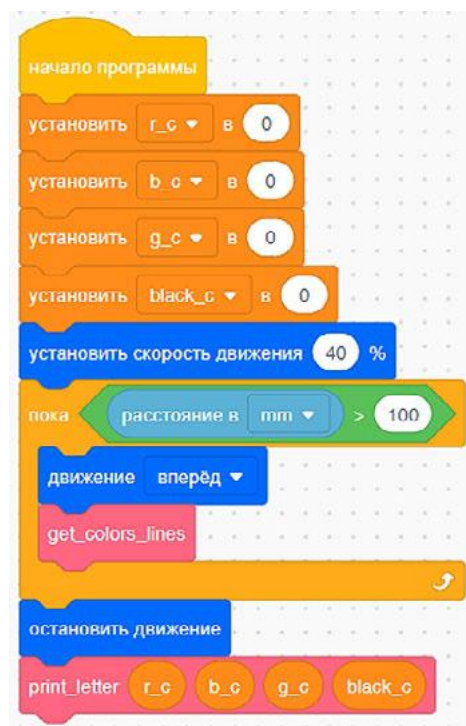
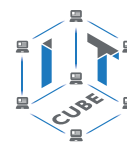


Рис. 81. Блок по подсчёту линий разных цветов

В данном блоке используются вложенные условные структуры, позволяющие определить чёрный цвет, который не задан явно в списке доступных цветов. Общий вид программы по подсчёту линий с выводом приведён на рисунке 82.





В результате выполнения этой программы в консоль будет выведена информация о текущей дорожке, на которой стоит VR-робот.

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание на платформе VEXcode VR.

Задание. Написать программу по подсчёту только красных и чёрных линий.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: блоки для взаимодействия с датчиками, алгоритм подсчёта линий нескольких цветов.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Каким образом можно кодировать сообщения с помощью линий?
- Какую переменную необходимо дополнительно ввести в программу, чтобы улучшить её?
- Как выглядит алгоритм подсчёта чёрных линий?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Написать программу по объезду VR-роботом всех дорожек за один проход.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 28

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 5. Творческий проект.

Тема урока: Создание собственного проекта с использованием максимально возможного количества датчиков.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с выполнением проектов на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: поставить задачу для реализации проекта по очистке кораллового рифа на платформе VEXcode VR (<http://vr.vex.com>).

Очистка кораллового рифа

На предыдущих занятиях были рассмотрены возможности среды программирования VEXcode VR. Теперь настало время использовать полученные вами знания для решения задач, приближённых к реальности. Одной из наиболее серьёзных экологических проблем на сегодняшний день является загрязнение Мирового океана отходами человеческой деятельности.

В проекте по очистке кораллового рифа в вашем распоряжении будет VR-робот, который поможет очистить мангровые заросли. Используя карту, соберите как можно больше мусора перед тем, как солнечные батареи иссякнут, и VR-робот выключится. Карта представляет собой симуляцию дна Мирового океана, содержащего на своей поверхности мусор (рис. 83).



Рис. 83. Карта по очистке кораллового рифа от мусора

Обратите внимание, что в нижнем левом углу карты появились два дополнительных индикатора, один из которых характеризует количество собранного мусора, а другой — уровень заряда VR-робота. Как вы понимаете, чем более медленный алгоритм будет использован при работе VR-робота, тем он быстрее разрядится и меньше соберёт мусора. Поэтому не забудьте в начале программы использовать блоки для установки максимальных значений скоростей и поворотов VR-робота (рис. 84).

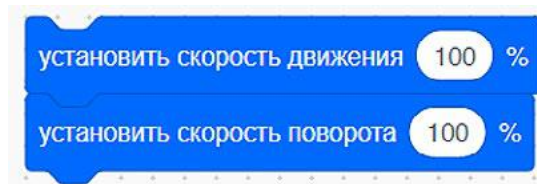
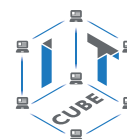


Рис. 84. Блоки установки скоростей

На уроке вам предлагается использовать для сбора мусора только команды управления трансмиссией без использования датчиков. Пример такой программы представлен на рисунке 85.

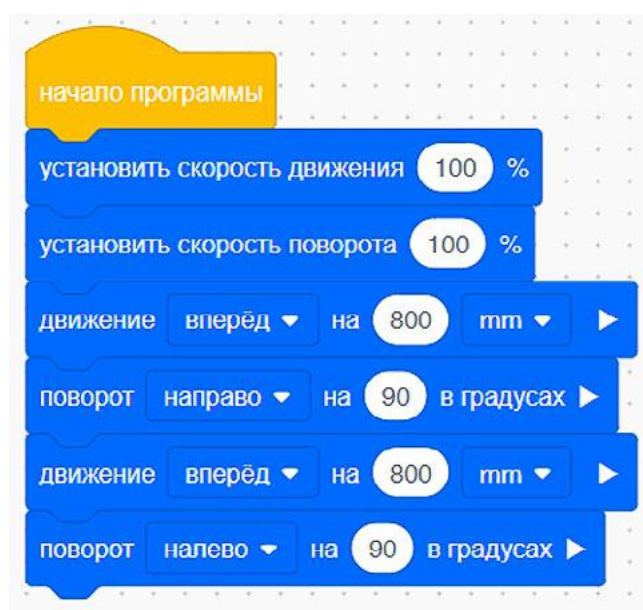


Рис. 85. Пример программы по очистке кораллового рифа

Помимо этого необходимо разработать стратегию сбора мусора, которая может вам пригодиться на следующих занятиях.

Деятельность учащихся: изучают описание проекта, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (20 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR.

Задание 1. Описать стратегию по сбору мусора.

Задание 2. Написать программу по сбору максимально возможного количества мусора.

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

Каким образом можно эффективнее всего собирать объекты в океане?

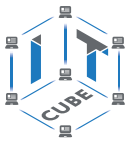
Какие технологии по очистке Мирового океана вам известны?

Как выглядит алгоритм поведения VR-робота для сбора мусора?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап IV. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.



Задание. Написать программу по сбору VR-роботом только центрального объекта и всех объектов слева.

Важно!

Используйте карту по очистке кораллового рифа от мусора.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап V. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXcode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 29

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 5. Творческий проект.

Тема урока: Создание собственного проекта с использованием максимально возможного количества датчиков.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с выполнением проектов на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

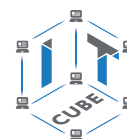
Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (10 мин)

Деятельность учителя: поставить задачу для реализации проекта по очистке кораллового рифа на платформе VEXcode VR (<http://vr.vex.com>).

Очистка кораллового рифа с использованием датчиков

На предыдущем занятии были рассмотрены возможности VR-робота для сбора мусора с поверхности дна океана без использования датчиков. Как вы понимаете, данный алгоритм не является оптимальным, так как не задействует уникальные возможности встроенных датчиков.

Опираясь на опыт, полученный в ходе изучения других карт VEXcode VR, можно утверждать, что использование датчиков значительно упрощает почти любую задачу. Благодаря датчикам, VR-робот способен реагировать на изменения в окружающей среде и со-



вершать определённые действия. Отметим, что после очистки всех базовых «мусорных» объектов карта динамически изменяется (рис. 86). Это существенно затрудняет дальнейшее проектирование алгоритмов с использованием только блоков движения.



Рис. 86. Динамическое появление новых объектов на карте

Для решения этой задачи необходимо использовать все доступные на VR-роботе датчики (расстояния, цвета), встроенный навигатор по координатам. Например, значение текущей координаты VR-робота позволит избежать столкновения с ценными мангровыми зарослями (рис. 87).

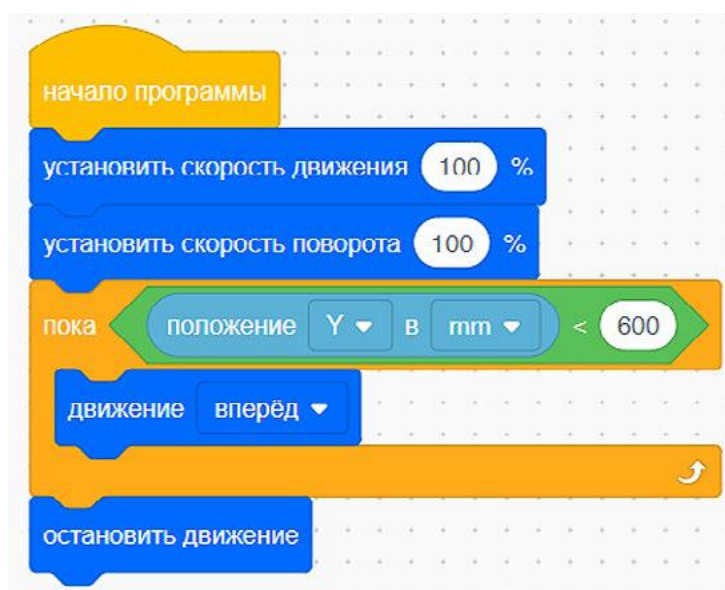


Рис. 87. Ограничение движения VR-робота по координатам

Задача — убрать как можно больше количества мусора. Результат будет отображён в вашем сертификате исследователя, который вы получите в конце испытания (рис. 88).



Рис. 88. Сертификат исследователя, выданный за очистку кораллового рифа

Деятельность учащихся: изучают описание проекта, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (25 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR.

Задание 1. Описать стратегию по сбору мусора.

Задание 2. Написать программу по сбору максимально возможного количества мусора с использованием датчиков.

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Какие блоки датчиков лучше всего использовать для сбора мусора?
- Как лучше всего комбинировать между собой датчики и блоки движения для решения этой задачи?
- Как выглядит алгоритм поведения VR-робота для сбора мусора с использованием датчиков?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап IV. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Постарайтесь улучшить результат, достигнутый вами на уроке.

Важно!

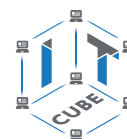
Используйте карту по очистке кораллового рифа от мусора.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап V. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.



Дидактические материалы

1. Платформа программирования роботов VEXcode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).
2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).
3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 30

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 5. Творческий проект.

Тема урока: Создание собственного проекта с использованием максимально возможного количества датчиков.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с выполнением проектов на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: поставить задачу для реализации проекта по декодированию числового кода на платформе VEXcode VR (<http://vr.vex.com>).

Изучение новых блоков для решения задачи декодирования

Помимо задач по очистке дна Мирового океана от мусора, разработчикам каждый день приходится решать задачи, связанные с защитой информации. Эта проблема является одной из самых актуальных в мире IT, так как множество сторонних лиц (например, киберпреступников) и организаций могут похитить ваши личные данные и использовать их в не самых благих целях.

Справочник!

Самым простым вариантом распознавания информации является преобразование чисел из разных систем счисления. Учитывая принцип работы электрических схем, самой распространённой и наиболее старой системой исчисления является двоичная (рис. 89).

8	7	6	5	4	3	2	1
128	64	32	16	8	4	2	1
0	1	0	1	1	0	0	1
В десятичной системе: 89							

Рис. 89. Число 89 в двоичной системе исчисления

Обратите внимание, что в первой строчке указано количество раз, которое необходимо перемножить число 2 для получения чисел, написанных во второй строчке.

Для выполнения проекта вам нужно будет использовать блок **Списка** (рис. 90).

Рис. 90. Создание блока **Списка**

Списки позволяют хранить данные одинаковой природы для более удобной организации информации и доступа к ней в ходе выполнения программы. Давайте рассмотрим альтернативу списку в виде последовательного создания переменных (рис. 91).

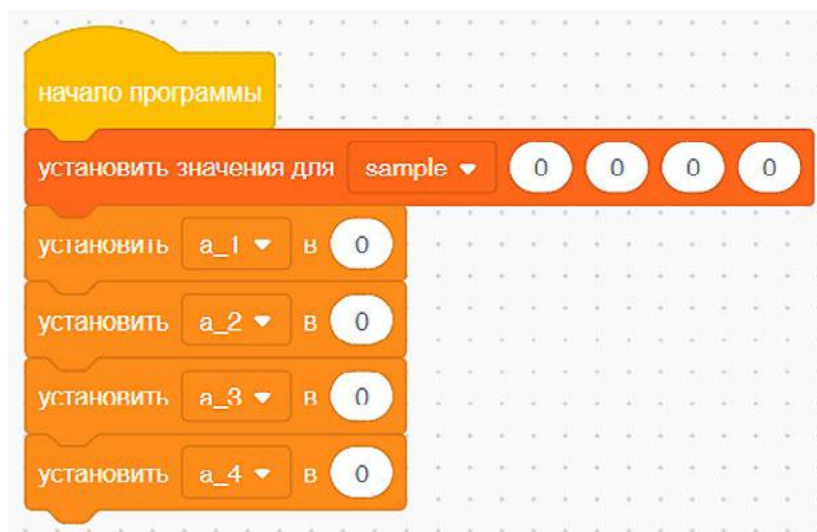


Рис. 91. Способы представления данных

Как вы могли заметить, список представляет собой более компактную конструкцию по сравнению с инициализацией нескольких переменных даже на этапе инициализации. Для обращения к элементам списка используют индексы, которые позволяют в каждый момент времени получать нужную информацию о необходимом элементе (рис. 92).



Рис. 92. Получение значения элемента по индексу

В отличие от классического программирования, нумерация элементов списка в VEXcode VR начинается с первого индекса. На рисунке 93 приведена программа, в которой используется список на примере печати информации в консоль вывода.

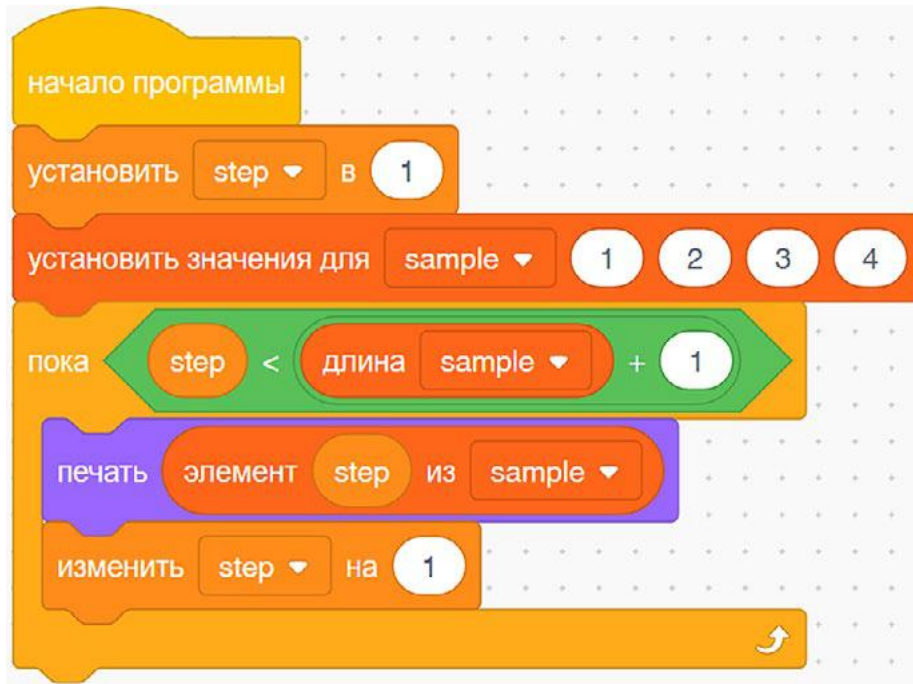
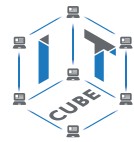


Рис. 93. Пример программы по использованию блока **Списка**

В данной программе снова используется понятие шага цикла для доступа к элементам **Списка**. Цикл закончится только после того, как значение шага достигнет максимального индекса в цикле. Обратите внимание на необходимость увеличения максимального значения элемента **Списка**, так как при отсутствии этой операции максимальный индекс не будет выведен в консоль.

Деятельность учащихся: изучают описание проекта, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание на платформе VEXcode VR.

Задание. Написать программу по последовательному присваиванию элементам **Списка** разных значений (с использованием цикла).

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: блок **Списка**, элементы **Списка**, работа с элементами **Списка**.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- В каких случаях может быть полезно использовать **Списки**?
- Как получить доступ к элементам **Списка**?
- Каким образом можно создать **Список** в VEXcode VR?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Напишите программу по подсчёту синих и зелёных линий на поле.

Важно!

Значения необходимо сохранить в блоке **Списка**.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXcode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 31

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 5. Творческий проект.

Тема урока: Создание собственного проекта с использованием максимально возможного количества датчиков.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с выполнением проектов на базе платформы VEXcode VR.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

128	0	0	0	0	0
64	1	1	1	0	0
32	0	0	0	0	0
16	1	0	1	0	0
8	0	0	1	0	0
4	1	1	0	0	0
2	1	0	0	0	0
1	0	1	0	0	0
A					

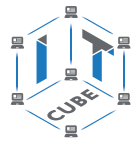
Рис. 94. Карта Декодирование сообщений

Деятельность учителя: поставить задачу для реализации проекта по декодированию числового кода на платформе VEXcode VR (<http://vr.vex.com>).

Изучение карты по декодированию объектов и постановка задачи

На предыдущем занятии были изучены понятия списков и двоичной системы. Теперь настало время применить полученные знания на практике. Рассмотрим карту по декодированию сообщения (рис. 94).

На данной карте показано несколько дорожек, каждая из которых содержит зашифрованное сообщение. Нашей задачей является использование всех возможных блоков и датчиков для реализации алгоритма по нахождению декодированного числа для каждой из дорожек. Резуль-



татом программы должна стать сумма всех декодированных чисел. Поэтому обратите особое внимание на подсчёт каждого значения, так как оно влияет на итоговый результат.

Теперь перейдём к описанию блока, который позволяет переводить число из двоичной в десятичную систему счисления (рис. 95).

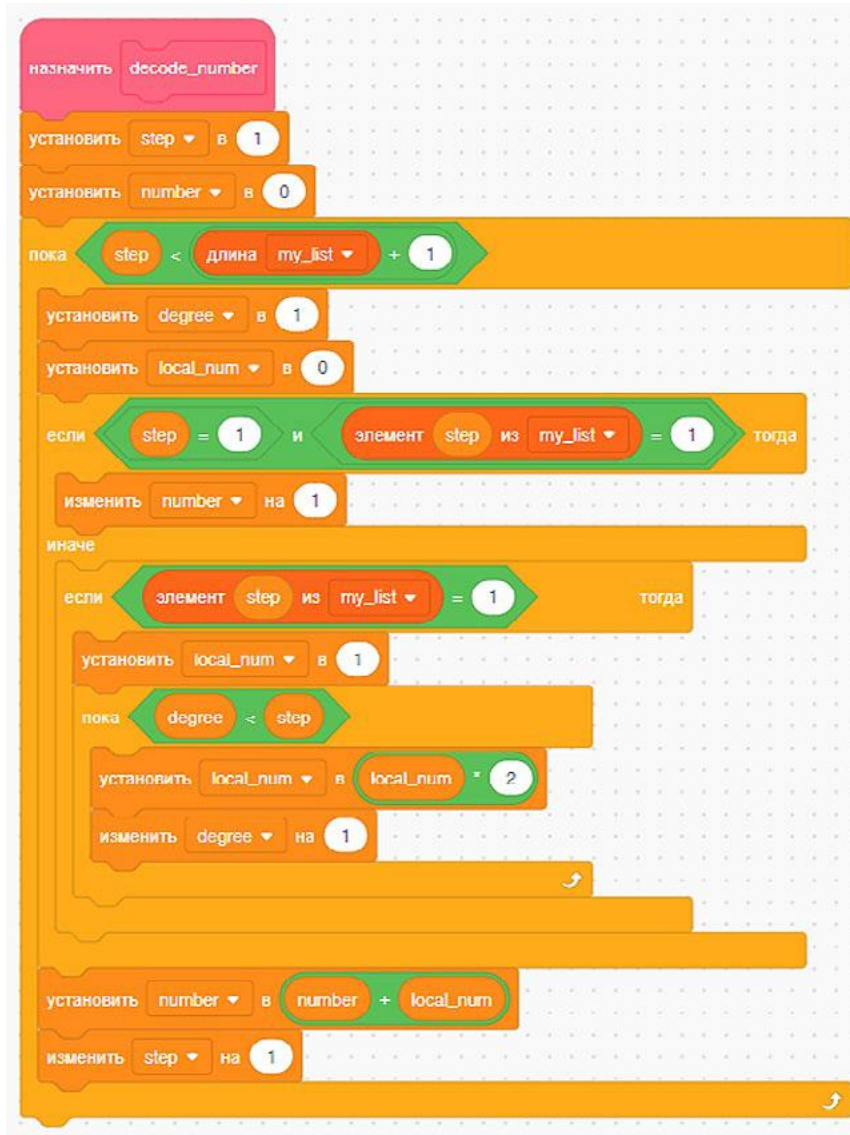


Рис. 95. Блок декодирования сообщения

Подробно обсудим шаги по декодированию, происходящие в рамках этого блока. В первую очередь создадим внешний цикл, который будет работать до тех пор, пока индекс списка не достигнет конечного значения. Рассмотрим случай, при котором первый разряд числа в двоичной системе будет равен 1, т. е. 2 в нулевой степени. Данный шаг необходим, так как в VEXcode VR нет встроенного инструмента возведения в степень и получение 2 в нулевой степени возможно только с помощью непосредственного назначения 1 в качестве результата. Вторая часть блока **Если...тогда...иначе** отвечает за возведение 2 в нужную степень, если значение в списке равно 1 .

Давайте посмотрим на блок-схему алгоритма, который будет описывать проезд VR-робота по одной из дорожек (рис. 96).

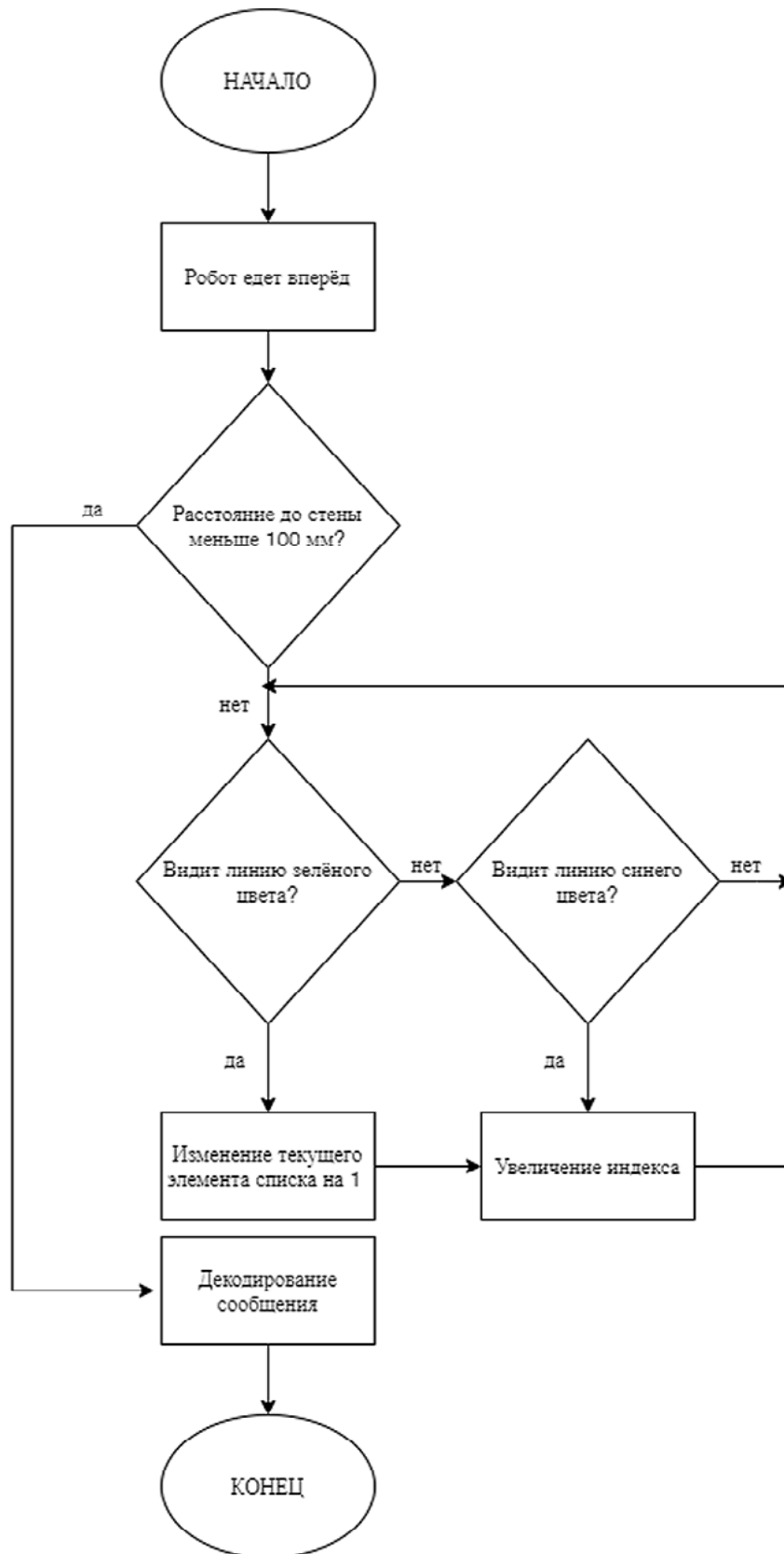
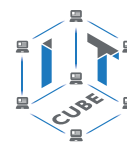


Рис. 96. Алгоритм чтения сообщения

Деятельность учащихся: изучают описание проекта, слушают объяснения учителя, задают вопросы.

**Этап II. Актуализация знаний и пробное учебное действие (25 мин)**

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе VEXcode VR.

Задание 1. Написать программу по объезду VR-роботом одной дорожки.

Задание 2. Написать программу по объезду VR-роботом трёх дорожек.

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Напишите программу по получению конечного результата (суммы всех декодированных чисел) на карте В.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап IV. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы

1. Платформа программирования роботов VEXCode VR [электронный ресурс] // URL: <https://vr.vex.com> (дата обращения: 15.04.2021).

2. Информатика. Уровень 1 «Блоки» [электронный ресурс] // URL: <https://education.vex.com/stemlabs/cs/computer-science-level-1-blocks> (дата обращения: 15.04.2021).

3. Официальный сайт среды программирования Scratch [электронный ресурс] // URL: <https://scratch.mit.edu/> (дата обращения: 15.04.2021).

Урок 32

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 6. Дальнейшее развитие.

Тема урока: Основы программирования роботов на языке Си.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с созданием программ на базе среды программирования RobotC.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока**Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)**

Деятельность учителя: продемонстрировать интерфейс программы RobotC.

Знакомство с интерфейсом программы RobotC

На уроке будет изучена среда программирования RobotC, которая позволит нам на следующих занятиях создавать программы для управления роботом. Для успешной работы с любой средой разработки необходимо знать её интерфейс и основные инструменты. Одним из наиболее востребованных приложений по программированию образовательных наборов является программа RobotC (рис. 97).

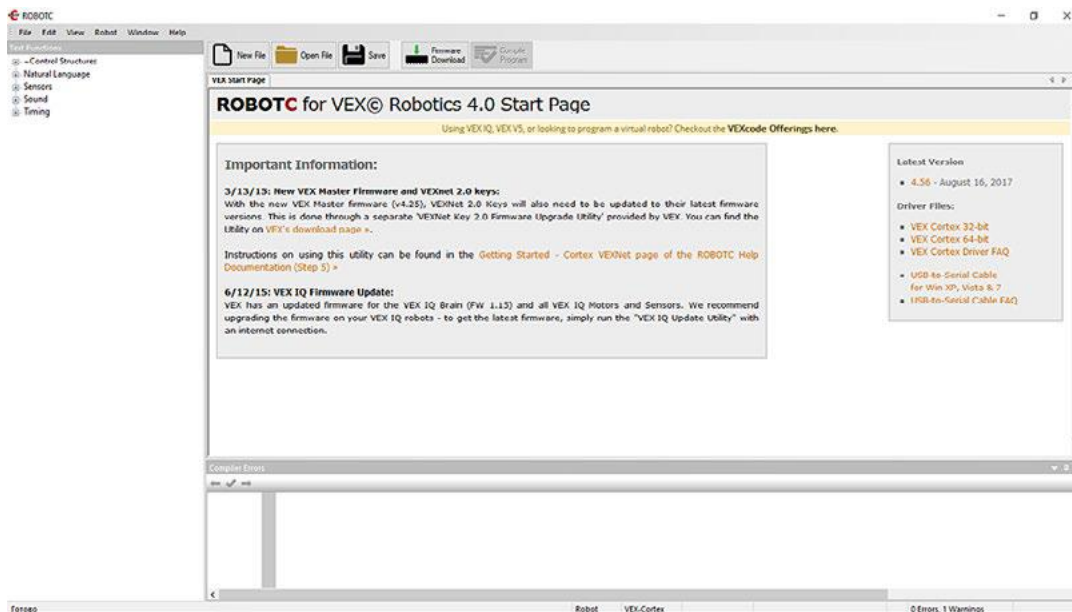


Рис. 97. Интерфейс программы RobotC

В самом верху интерфейса этой программы находится панель управления средой разработки, немного ниже представлена панель управления окном разработки программы. Кнопка **New File** позволяет создать новую программу, кнопка **Save** сохраняет программу, кнопка **Firmware Download** обновляет «мозги робота». Кнопка **Compile** преобразует программу в язык, понятный роботу, а кнопка **Download to Robot** загружает в робота данную программу.

Рассмотрим меню **Edit** на панели управления средой программирования RobotC (рис. 98).

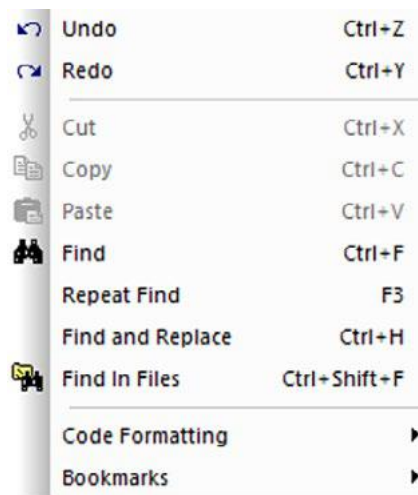


Рис. 98. Меню настройки **Edit**

В данном меню нас будут интересовать только три кнопки: **Undo** и **Redo**, которые осуществляют переход к предыдущему или следующему шагу соответственно, кнопка **Find** позволяет найти какие-либо строки кода. Кроме того, нас будет интересовать папка **Open Sample Program** в меню **File**, с помощью которой можно открыть примеры существующих программ в случае, если у вас возникли трудности при написании собственной программы. Наиболее важным меню для нас является меню **Robot** (рис. 99), которое позволит настроить нашего робота.

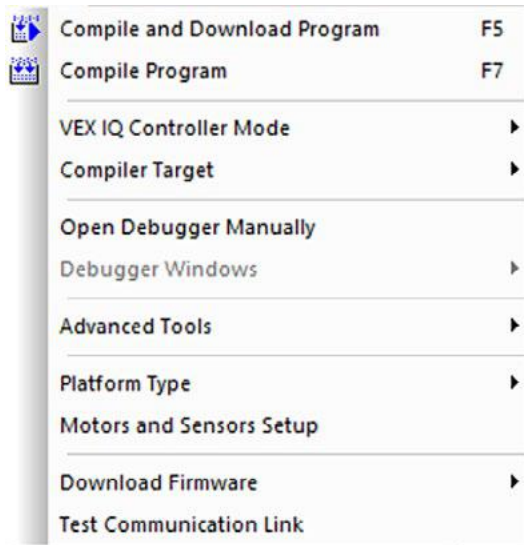
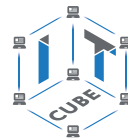


Рис. 99. Меню Robot

Первые два пункта были изучены ранее. Вкладка **VEX IQ Controller Mode** позволяет переключаться между управляемым и автономным режимами программирования робота. Во вкладке **Compile Target** доступна только одна кнопка **Physical Robot**. Вкладки с включённым словом debug являются более продвинутыми инструментами работы с кодом и редко применяются при программировании физической модели.

В разделе **Advanced Tools** нас будет интересовать кнопка **VEX IQ Joystick Viewer**. Она позволяет проверить работоспособность пульта управления. Перед началом программирования вам обязательно нужно выбрать тип вашей платформы в меню **Platform Type**, иначе возникнет ситуация, при которой вы и робот будете общаться абсолютно на разных языках. После того как будет создана новая программа, откроется возможность настройки устройств робота. Теперь можно переходить в меню настройки **Motors and Sensors Setup** (рис. 100).

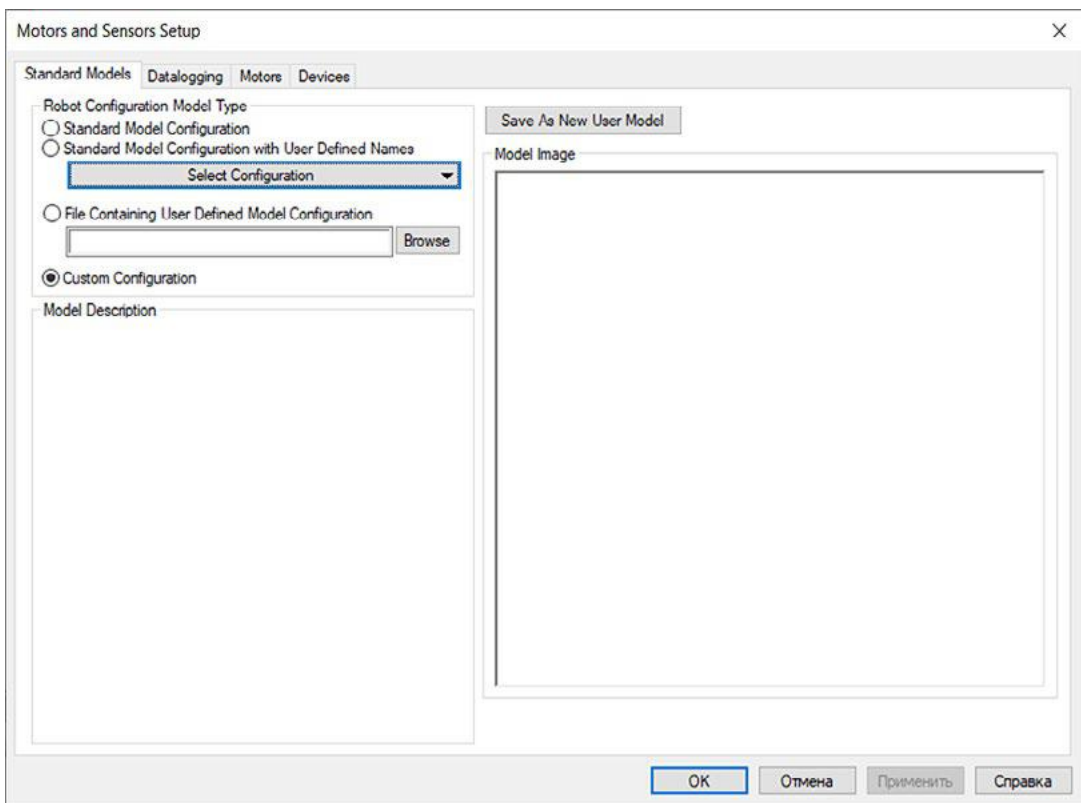
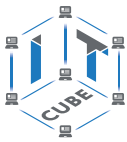


Рис. 100. Меню настройки Motors and Sensors Setup



Первая вкладка **Standard Models** позволяет вам выбрать автоматическую настройку вашего робота из списка. В этом списке уже есть наша модель **Autopilot Robot**, а также много других интересных моделей. Перейдём теперь ко вкладке **Motors**. Перед нами окажется несколько колонок. Первая колонка представляет собой список всех доступных портов подключения для моторов. Вторая колонка задаёт вашему порту управления уникальное имя, которое будет отображать функцию того или иного мотора. В третьей колонке можно выбрать тип мотора: в нашем случае выбор всего один. В четвёртой колонке можно изменить направление вращения мотора на противоположное. И наконец, пятая колонка определяет сторону расположения мотора (для мобильных роботов).

Аналогичные колонки, помимо четвёртой и пятой, приведены и во вкладке **Devices**. Единственным отличием будет список типов датчиков. Всего наборы содержат пять видов датчиков: цвета, касания, тактильный, расстояния и гироскоп.

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание в среде разработки RobotC.

Задание. Произвести первоначальную настройку среды RobotC и робота с помощью меню **Motors and Sensors Setup**.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: элементы интерфейса среды программирования RobotC, меню **Edit, Robot**.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Зачем нужно меню **File**?
- Для чего используются пункты в меню **Robot**?
- Каким образом можно настроить стандартную конфигурацию робота в среде разработки RobotC?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Подготовьте стандартную настройку робота и объясните необходимость использования всех подключённых устройств.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

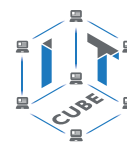
Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактический материал

Портал обучения «VEX Академия» [электронный ресурс] // URL: <http://vexacademy.ru/> (дата обращения: 15.04.2021).



Урок 33

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 6. Дальнейшее развитие.

Тема урока: Основы программирования роботов на языке Си.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с созданием программ на базе среды программирования RobotC.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: продемонстрировать использование типов данных в совокупности с математическими и логическими операциями в среде разработки RobotC.

Знакомство с типами данных

В программировании переменные невероятно важны, так как позволяют хранить информацию и использовать её в дальнейшем. Сначала может быть не совсем понятно, зачем вообще что-то записывать в переменные, если можно просто оперировать значениями без них. Понимание переменных придёт немного позже, когда вы будете работать с более сложными программами и вам потребуется хранить информацию в каком-либо месте.

Типы переменных в языке Си указываются перед названием переменной. От конкретного типа данных будет зависеть содержимое переменной. Если мы укажем переменную с типом данных для целых чисел `int`, то в неё нельзя будет записать строку или число с точкой.

При составлении программы можно записывать новые значения в переменную, но тип данных должен оставаться неизменным:

```
float some_value = 2.95;  
some_value = 6.9;
```

Справочник!

Типы данных для создания переменных

Целые числа

- **Short** — предназначен для хранения целых чисел. Диапазон чисел от $-32\,768$ до $32\,767$.
- **Unsigned short** — предназначен для хранения целых положительных чисел. Диапазон чисел от 0 до $65\,535$.
- **Int** — предназначен для хранения целых чисел. Диапазон чисел от $-2\,147\,483\,648$ до $2\,147\,483\,647$.
- **Unsigned int** — предназначен для хранения целых положительных чисел. Диапазон чисел от 0 до $4\,294\,967\,295$.
- **Long** — предназначен для хранения целых чисел. Диапазон чисел от $-9\,223\,372\,036\,854\,775\,808$ до $9\,223\,372\,036\,854\,775\,807$.
- **Unsigned long** — предназначен для хранения целых положительных чисел. Диапазон чисел от 0 до $18\,446\,744\,073\,709\,551\,615$.

Разница между типами заключается только в диапазоне чисел, который можно записать в переменную.

Не используйте большие типы данных по типу `long`, если число маленькое, так как чем больше диапазон, тем больше памяти требуется компьютеру на конкретную переменную.

Числа с точкой

- **Float** — для создания чисел с плавающей точкой. Диапазон чисел от от $-3.4 \cdot 10^{38}$ до $3.4 \cdot 10^{38}$.
- **Double** — для создания чисел с плавающей точкой. Диапазон чисел от $\pm 4.9 \cdot 10^{-324}$ до $\pm 1.8 \cdot 10^{308}$.

Прочие типы данных

- **Bool** — логический тип данных, который предназначен для записи истинного (`true`) или ложного (`false`) значения.
- **Char** — тип данных для работы с символами, позволяющий поместить в одинарных кавычках какой-либо символ.

Для создания переменной необходимо указать тип данных и назвать её. Также можно сразу же добавить к ней значение:

```
int a = 0; // Добавление значения сразу
```

```
float c; // Создание переменной без значения
```

```
short b, y = 342; // Создание нескольких переменных
```

Для вызова переменной вы можете обратиться к ней по имени.

Особенности программирования в среде RobotC

Среда программирования RobotC предназначена для работы с робототехническими устройствами, поэтому, в отличие от классических сред программирования, здесь описывается работа с процессами.

Основной блок программы выглядит следующим образом:

```
task main()  
{  
  //Основная программа  
}
```

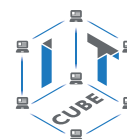
Слово `task` указывает компилятору на место начала выполнения процесса, а слово `main` показывает, что данный процесс имеет наивысший приоритет выполнения. Фигурные скобки являются ограничителями выполнения для блока и показывают компилятору места его начала и конца.

Давайте рассмотрим пример программы по сложению двух чисел целого типа данных:

```
task main()  
{  
  int a;  
  int b;  
  a = 2;  
  b = 2;  
  a = a + b;  
}
```

Обратите внимание на точку с запятой, которую необходимо ставить в конце каждой строки!

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

**Этап II. Актуализация знаний и пробное учебное действие (15 мин)**

Деятельность учителя: предлагает учащимся под его контролем выполнить задание в среде разработки RobotC.

Задание. Произвести математические операции сложения, вычитания, умножения и деления между переменными разных типов данных.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: типы данных, особенности среды программирования RobotC.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

Какие целые типы данных вы знаете?

Какие типы данных для чисел с плавающей точкой вам известны?

Как выглядит основной процесс в среде RobotC?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Напишите программу, которая будет решать уравнение вида $x = a * b + c * d$, где все значения справа задаются с клавиатуры.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактические материалы:

1. Сайт itProger [электронный ресурс] // URL: <https://itproger.com/course/c-programming/2> (дата обращения: 15.04.2021).

2. Портал обучения «VEX Академия» [электронный ресурс] // URL: <http://vexacademy.ru/> (дата обращения: 15.04.2021).

Урок 34

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 6. Дальнейшее развитие.

Тема урока: Основы программирования роботов на языке Си.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с созданием программ на базе среды программирования RobotC.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: продемонстрировать использование модуля отладки в среде разработки RobotC.

Модуль отладки программы Debug

Прежде чем перейти к изучению конструкций ветвлений и циклов, необходимо научиться просматривать выходные данные, возникающие в ходе выполнения программы. В среде программирования RobotC предусмотрена такая возможность посредством встроенного отладчика, доступ к которому можно получить, выбрав в меню **Robot** пункт **PC-Based Emulator** (рис. 101).

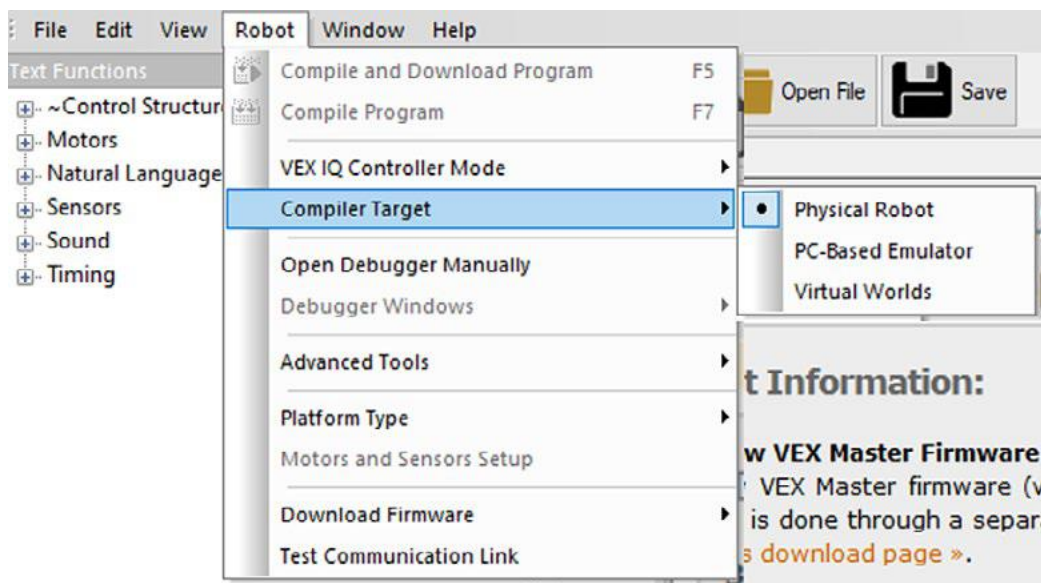


Рис. 101. Выбор параметра загрузки программы

Данный пункт позволяет запускать программу в режиме реального времени без использования робота. В качестве примера рассмотрим следующую программу:

```
task main()
{
    int a;
    a = 2;
    a = a + 2;
}
```

Для запуска процесса отладки необходимо нажать на кнопку **Download to robot**. Перед вами откроется окно отладки (рис. 102).

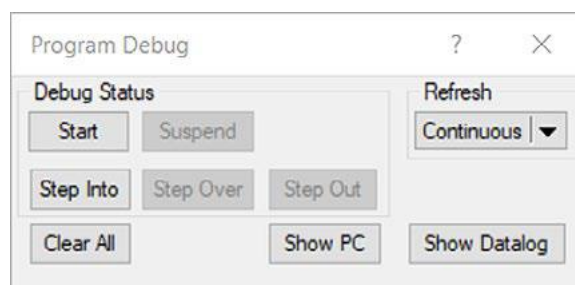
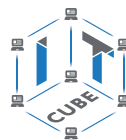


Рис. 102. Окно отладки Debug



После появления окна отладки необходимо нажать на кнопку **Step Out** для доступа к первому шагу процесса отладки. Первая строка вашей программы подсветится жёлтым цветом (рис. 103).

```

1
2
3  ⇒ task main()
4     {
5         int a;
6
7         a = 2;
8

```

Рис. 103. Исполняемая строка во время отладки программы

Для перехода к следующему шагу отладки необходимо снова нажать на кнопку **Step Out**. Часть шагов программы будет скрыта, так как они находятся в процессе сборки, связанном с настройкой контроллера и процесса **task main**, поэтому для доступа к следующей строке нужно нажать на кнопку **Step Out** несколько раз. Все выходные данные будут отображаться внизу экрана в консоли с названием **Local Variables** (рис. 104).

Local Variables		
Offset	Variable	Value
0x0000	a	4

Global Variables Local Variables VEX IQ Device Driver Stats

Рис. 104. Консоль Local Variables

Отладка программы необходима не только для вывода значений, но также для поиска ошибок и нестыковок в коде. Отметим, что существует отдельный раздел программирования, который отвечает за поиск и написание автоматических тестов для поиска и выявления возможных ошибок.

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание в среде разработки RobotC.

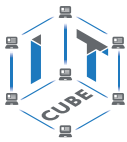
Задание. Воспроизвести задачу из домашнего задания предыдущего урока и убедиться в том, что выходные данные совпадают с ожидаемыми.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: отладка программ, консоли вывода.

Деятельность учащихся: демонстрируют, как усвоили материал.



Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Зачем нужна пошаговая отладка программы?
- Как в RobotC реализована пошаговая отладка программы?
- Почему первый шаг отладки долго выполняется?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Найдите ошибки в программе:

```
task main()  
{  
    int a;  
    int b;  
    b = a;  
    a = 2  
    a = a +- 2;  
}
```

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактический материал

Портал обучения «VEX Академия» [электронный ресурс] // URL: <http://vexacademy.ru/> (дата обращения: 15.04.2021).

Урок 35

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 6. Дальнейшее развитие.

Тема урока: Основы программирования роботов на языке Си.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с созданием программ на базе среды программирования RobotC.

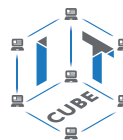
Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока

Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (15 мин)

Деятельность учителя: продемонстрировать использование ветвлений и циклов в среде RobotC.



Ветвление в RobotC

Как и большинство языков программирования, RobotC позволяет использовать условные операторы и циклы для создания многовариантности программ и создания автономных мобильных роботов.

Условный оператор **if...else** выглядит следующим образом:

```
if (условие)
{
//Блок if
}
else
{
//Блок else
}
```

Рассмотрим условный оператор **if...else**. Если условие в скобках после *if* будет правдивым, то будет выполнен блок после *if*, если условие будет ложным, то — блок после *else*.

Рассмотрим пример программы, которая продемонстрирует принцип работы данного условного оператора:

```
task main()
{
    int a;
    int b;
    a = 5;
    b = 0;
    if (b != 0)
    {
        a = a / b;
    }
    else
    {
        a = 0;
    }
}
```

В рассматриваемом примере условный оператор позволяет избежать ошибки деления на ноль, которая обязательно появится во время выполнения программы. Так как в нашем случае значение переменной *b* будет равно нулю, то переменная *a* также обратится в нуль в соответствии с блоком после *else* (рис. 105).

Offset	Variable	Value
0x0000	a	0
0x0004	b	0

Global Variables Local Variables VEX IQ Device Driver Stats

Рис. 105. Результат работы программы

Циклы в RobotC

На практике иногда бывает необходима многократная проверка условия для достижения результата. Для этого можно использовать несколько блоков *if*, но это приведёт к усложнению программного кода и ухудшит читаемость программы. В связи с этим используются циклы.

Одним из наиболее часто используемых циклов является цикл **while**:

```
while (условие)
{
  //Блок цикла
}
```

Давайте рассмотрим пример программы с использованием цикла **while**:

```
task main()
```

```
{
  int a;
  int b;

  a = 0;
  b = 0;
  while (a < 4)
  {
    a = a + 1;
    b = b + 2;
  }
}
```

В данной программе цикл будет выполняться до тех пор, пока переменная *a* не достигнет значения 4. После этого цикл закончится и в переменных останутся значения, которые были сохранены на последнем шаге цикла (рис. 106).

Offset	Variable	Value
0x0000	a	4
0x0004	b	8

Рис. 106. Результат работы программы

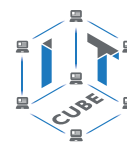
Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (15 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить задание в среде разработки RobotC.

Задание. Написать программу по решению линейного уравнения вида $y = ax + b$, где *y*, *a* и *b* задаются заранее.

Деятельность учащихся: самостоятельно выполняют предложенное задание.

**Этап III. Повторение нового материала (5 мин)**

Деятельность учителя: кратко повторяет новый материал: ветвления, циклы.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Как выглядит ветвление в среде программирования RobotC?
- Как выглядит цикл в среде программирования RobotC?
- Чем различаются циклы и ветвления?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Напишите программу для последовательного вывода в консоль значений таблицы умножения до десяти для числа, заданного заранее.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактический материал

Портал обучения «VEX Академия» [электронный ресурс] // URL: <http://vexacademy.ru/> (дата обращения: 15.04.2021).

Урок 36

Уровень образования: основное общее.

Предмет: информатика (раздел «Робототехника»).

Класс: 5–7.

Уровень изучения (базовый): Модуль 6. Дальнейшее развитие.

Тема урока: Основы программирования роботов на языке Си.

Тип урока: комбинированный.

Цель урока: ознакомить учащихся с созданием программ на базе среды программирования RobotC.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в Интернет.

Ход урока**Этап I. Постановка цели и задач урока, мотивация к учебной деятельности (10 мин)**

Деятельность учителя: продемонстрировать использование функций движения робота в среде RobotC.

Команды движения робота

На данном уроке будут рассмотрены основные команды управления движениями робота на примере платформы VEX IQ. В первую очередь необходимо настроить среду под нашего робота. Для этого в меню настройки **Motors and Sensors Setup** необходимо

выбрать стандартную конфигурацию робота **Autopilot Robot** для автоматической настройки всех портов подключения (рис. 107).

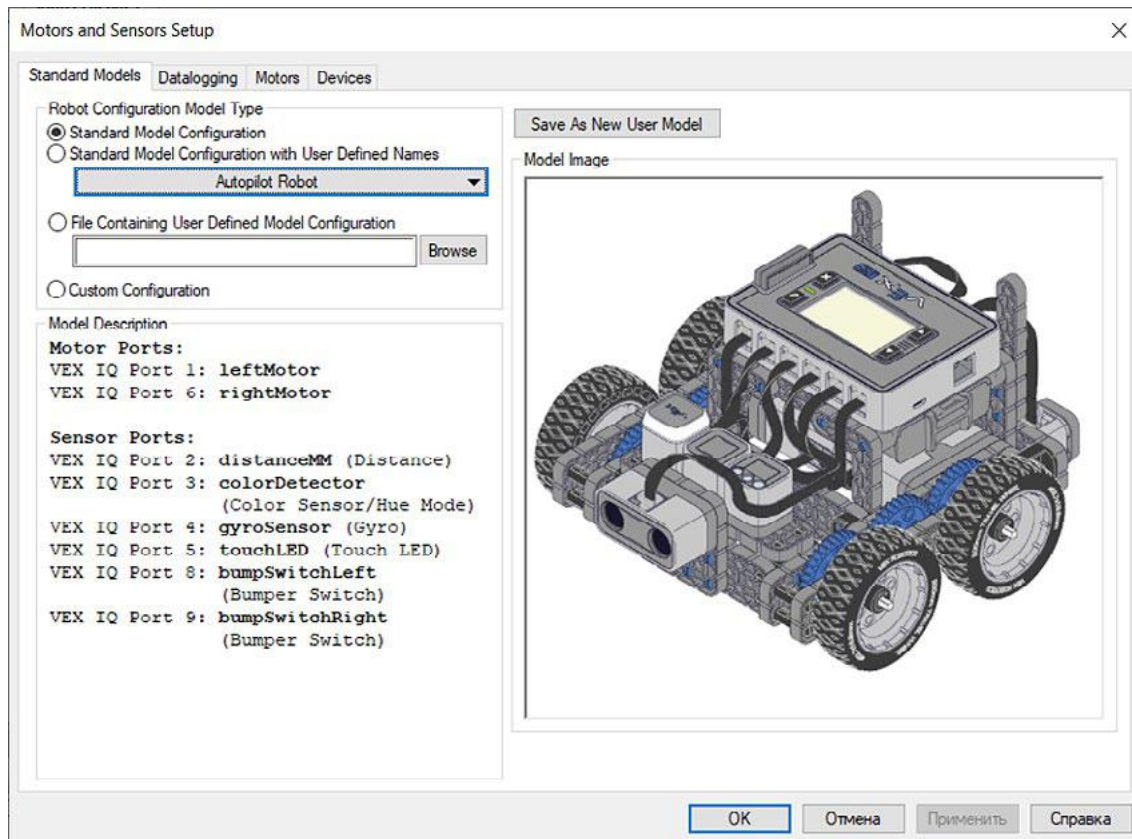


Рис. 107. Настройка стандартной конфигурации робота

Рассмотрим следующую программу:

```
#pragma config(StandardModel, "Autopilot Robot")
```

```
task main()
{
    setMotor(leftMotor, 50);
    setMotor(rightMotor, 50);
    wait(1, seconds);
    stopAllMotors();
}
```

Первая строка программы генерируется автоматически после настройки в меню **Motors and Sensors Setup**. После объявления уже знакомого процесса *task main* необходимо активировать приводы робота. Для этого используется команда:

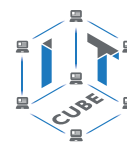
```
setMotor(leftMotor, 50);
```

В скобках после вызова команды через запятую указываются название порта (или номер порта) подключения привода и скорость (в диапазоне от 1 до 127) соответственно. Использование минуса перед значением скорости позволяет изменять направление движения приводов.

После активации приводов используется команда:

```
wait(1, seconds);
```

Данная команда позволяет удержать действия с целью проезда робота в течение некоторого времени. Через запятую указываются значение временного промежутка и единица измерения времени.



После того как робот проедет вперёд указанное время, необходимо остановить работу всех используемых приводов с помощью команды:

```
stopAllMotors();
```

Деятельность учащихся: изучают новый материал, слушают объяснения учителя, задают вопросы.

Этап II. Актуализация знаний и пробное учебное действие (20 мин)

Деятельность учителя: предлагает учащимся под его контролем выполнить некоторые задания на платформе RobotC.

Задание 1. Написать программу по отъезду робота назад.

Задание 2. Написать программу по повороту робота направо.

Деятельность учащихся: самостоятельно выполняют предложенные задания.

Этап III. Повторение нового материала (5 мин)

Деятельность учителя: кратко повторяет новый материал: команды движения, остановки, таймер.

Деятельность учащихся: демонстрируют, как усвоили материал.

Этап IV. Проверка понимания и первичное закрепление (5 мин)

Деятельность учителя: задаёт учащимся следующие вопросы.

- Что указывается в скобках команды **setMotor**?
- Что указывается в скобках команды **stopAllMotors**?
- Зачем нужна команда **wait**?

Деятельность учащихся: отвечают на предложенные вопросы.

Этап V. Информация о домашнем задании, инструктаж по его выполнению (3 мин)

Деятельность учителя: сообщает учащимся домашнее задание.

Задание. Напишите программу по объезду роботом квадрата.

Деятельность учащихся: задают уточняющие вопросы о выполнении домашнего задания.

Этап VI. Рефлексия деятельности на уроке (2 мин)

Деятельность учителя: спрашивает учащихся об их впечатлениях от урока, что понравилось, что осталось непонятным.

Деятельность учащихся: отвечают на вопросы учителя.

Дидактический материал

Портал обучения «VEX Академия» [электронный ресурс] // URL: <http://vexacademy.ru/> (дата обращения: 15.04.2021).

Приложение

Регламенты соревнований для роботов

В настоящее время мир бросает нам множество вызовов, связанных с появлением новых проблем. Одной из наиболее широко распространённых и глобальных проблем является задача по очистке различных сред обитания от мусора, созданного человеком. Для её решения было разработано множество решений, одним из которых является сортировка мусора для его последующей переработки.

Цель: разработать конструкцию робота, которая будет наиболее эффективно справляться с задачей сортировки мусора, и алгоритм, который позволит выполнять её за наименьшее время.

Задачи:

- 1) разработка каркаса робота;
- 2) создание захватного устройства;
- 3) разработка алгоритмов сбора и сортировки мусора.

Требуемое оборудование:

- 1) робототехнический набор с металлическими или пластмассовыми компонентами для сборки робота;
- 2) программируемый контроллер;
- 3) датчики, совместимые с программируемым контроллером;
- 4) ПК с установленным ПО, совместимым с контроллером;
- 5) кубики разных цветов для имитации мусора различных категорий (рис. 108).

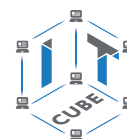


Рис. 108. Пример кубика для проведения соревнований

Для проведения соревнований рекомендуется использовать ограниченное пространство, которое будет имитировать сортировочный цех (рис. 109).



Рис. 109. Пример огороженного периметра для проведения соревнований



Система начисления баллов

Участникам соревнования необходимо доставить кубики в зоны соответствующих цветов. Важно помнить о том, что кубики, находящиеся в башнях, необходимо разгружать по очереди и не допускать падения башен.

Перед началом соревнований судейская коллегия определяет положение всех элементов поля и выдаёт схему их расположения (рис. 110) участникам.

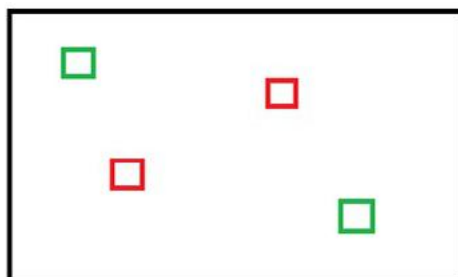


Рис. 110. Пример расположения элементов в начале соревнований

Рекомендуется начислять баллы по следующему принципу.

Куб доставлен в свою зону — 1 балл.

Все кубы одного цвета доставлены в соответствующую зону — 3 балла.

Штраф за упавшую башню — 2 балла.

Башня безопасно разгружена — 2 балла.

Все башни безопасно разгружены — 6 баллов.

Тот участник соревнования, который наберёт наибольшее количество баллов, объявляется победителем.

Курносенко Михаил Валерьевич
Мацаль Игнатий Игнатьевич

Реализация дополнительной общеобразовательной программы
по тематическому направлению «Программирование роботов» с использованием обо-
рудования центра цифрового образования детей «IT-куб»

Методическое пособие



Под редакцией С. Г. Григорьева

Центр Естественно-научного и математического образования
Руководитель Центра *З. Г. Гапонюк*
Ответственный за выпуск *В. В. Кудрявцев*
Редактор *В. В. Кудрявцев*

Художественное оформление *Т. В. Глушкова*
Компьютерная вёрстка и техническое редактирование *Т. М. Дородных*
Корректор *Н. А. Смирнова*