



Частное образовательное учреждение дополнительного
профессионального образования
Донской экономико-правовой колледж
предпринимательства
(ДЭПК)

Методические указания

**по теме «Методика проведения лабораторных работ по
робототехнике в курсе информатики с использованием
оборудования центров «Точка роста»**

для слушателей курсов повышения квалификации и профессиональной
переподготовки по направлениям по направлениям
ПК.0150.0000 «Педагогика и организация общего образования»
**ПК.0180.0000 «Педагогика и организация дополнительного
образования»**

Очной, очно-заочной и заочной форм обучения

**Электронное издание локального внутриколледжного
распространения**

Методические указания составлены для слушателей, обучающихся по дополнительным образовательным программам повышения квалификации и профессиональной переподготовки по направлениям ПК.0150.0000 **«Педагогика и организация общего образования»**, ПК.0180.0000 **«Педагогика и организация дополнительного образования»**.

Методические указания обсуждены, одобрены и рекомендованы к изданию цикловой методической комиссией ДЭПК.

Присвоить методическим указаниям гриф *«Электронное издание локального внутрикледжного распространения»*.

Протокол № 7 от « 4 » мая 2021 г.

Председатель ЦМК  к.э.н., Горцевская Е.А.
(роспись)

Робототехника

Планируемые результаты освоения учебного предмета с описанием универсальных учебных действий, достигаемых обучающимися

Личностные:

- формирование профессионального самоопределения, ознакомление с миром профессий, связанных с информационными и коммуникационными технологиями;
- формирование умения работать в команде;
- развитие внимательности, настойчивости, целеустремлённости, умения преодолевать трудности;
- формирование навыков анализа и самоанализа.

Предметные:

- формирование понятий о различных компонентах робота и платформы VEXcode VR (программные блоки по разделам, исполнительные устройства, кнопки управления и т. д.);
- формирование основных приёмов составления программ на платформе VEXcode VR;
- формирование алгоритмического и логического стилей мышления;
- формирование понятий об основных конструкциях программирования: условный оператор if/else, цикл while, понятие шага цикла.

Метапредметные:

- 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;
- использование средств информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач;
- формирование информационной культуры;
- формирование умения аргументировать свою точку зрения на выбор способов решения поставленной задачи.

Формы контроля

Во время проведения курса предполагается текущий, промежуточный и итоговый контроль.

Текущий контроль проводится на каждом занятии с целью выявления правильности применения теоретических знаний на практике. Текущий контроль может быть реализован посредством следующих форм: наблюдение, индивидуальные беседы, тестирование, творческие работы, проблемные (ситуативные) задачи, практические работы, контрольные вопросы и т. д.

Примеры ситуативных задач по модулю 1

Задача 1. Петя запустил робота, который движется по следующей программе:

- 1) стартует с точки *A* и едет на запад со скоростью $V = 3$ м/мин в течение 60 с;
- 2) поворачивает на юг и столько же времени движется с удвоенной скоростью $2V$;
- 3) поворачивает на восток и едет с утроенной скоростью $3V$ такое же время, что на первых двух участках вместе взятых;

4) поворачивает на север и, проехав 6 м за 1,5 мин, добирается до финиша, расположенного в точке В.

Вопросы:

1. Какова длина первого участка пути? Ответ дайте в метрах с точностью до целых.
2. С какой постоянной скоростью на всём пути должен двигаться робот, чтобы проехать его за то же время? Ответ укажите в метрах в секунду с точностью до сотых.
3. Найдите расстояние между точкой старта А и точкой финиша В робота. Ответ дайте в метрах с точностью до целых.

Задача 2. Три колёсных робота А1, А2 и А3 одинаковой конструкции должны по очереди пройти лабиринт, двигаясь от входа (синий квадрат) к выходу (зелёный квадрат). Робот А1 содержит в памяти карту лабиринта, на которой отмечены синий и зелёный квадраты и указаны все стенки. Робот А2 не знает карты лабиринта и запрограммирован обходить его по правилу правой руки. Робот А3 не знает карты лабиринта и запрограммирован обходить его по правилу левой руки. Какой из роботов пройдёт лабиринт медленнее всего?

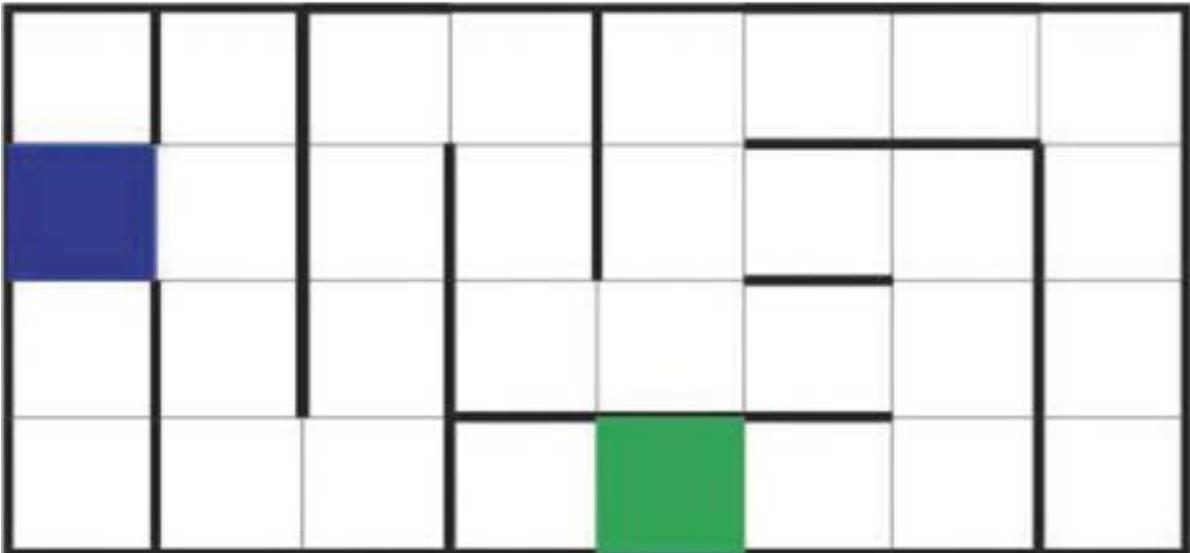


Рис. 2. Вид лабиринта

Промежуточный контроль проводится в рамках промежуточной аттестации после изучения нескольких модулей в виде подготовки и защиты творческих (проектных) работ, соревнований и состязаний.

Пример соревнования «Динамический лабиринт»

Цель: запрограммировать робота на решение лабиринта (прибытие на красный квадрат), в кратчайшие сроки.

Команды состоят только из одного участника.

Правила и подсчёт очков:

- 1) задача состоит в том, чтобы пройти лабиринт в кратчайшие сроки. Лабиринт считается пройденным, когда все колёса робота касаются красного квадрата;
- 2) максимальное время — 180 секунд. Если робот не завершил лабиринт за этот промежуток времени, время будет считаться как 200 секунд;
- 3) победителем становится команда с лучшим средним временем прохождения лабиринта из двух попыток. Если есть ничья, то в качестве тай-брейка используется лучшее время команды.

Тест по теме «Робот. Базовые понятия»

1. В каком году появилось слово РОБОТ?
А) 1920
Б) 1925
В) 1930
Г) 1935
2. Слово «Робот» — произошёл от чешского слова, которое означает...
А) RoboTech
Б) Robot
В) RobotLand
Г) *Robota*
3. Кто придумал три закона робототехники?
А) Валли
Б) А. Азимов
В) Г. Галилей
Г) К. Чапек
4. С 1968 г. «столицей роботов» считается
А) Китай
Б) Россия
В) *Япония*
Г) США
5. Как называется разработанный Aldebaran Robotics человекоподобный робот, поступивший в массовую продажу?
А) T-800
Б) Atlas
В) *Pepper*
Г) ASIMO

При проведении итоговой аттестации в форме проектной работы задание ориентировано на индивидуальное исполнение. Защита итогового проекта проходит в форме представления обучающимся технического задания на проект, работающего кода, ответов на вопросы преподавателя, обсуждения с учащимися достоинств и недостатков проекта.

Модуль 1. «Платформа 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, ракурсы 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.

Урок 1. Робот. Базовые понятия

Урок 2. Знакомство со средой VEXcode VR

Урок 3. Исполнительные механизмы конструкторов VEX

Лабораторная работа 1. Создание простейших программ (скриптов)

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.

Урок 4. Программируемый контроллер

Урок 5. Основные блоки

Лабораторные работы 2-3. Программирование блоков управления роботом

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».

Урок 6. Датчик местоположения, направление движения

Урок 7. Датчики цвета

Урок 8. Датчик расстояния

Урок 9. Управление магнитом

Лабораторная работа 4. Скрипты с датчиком местоположения

Лабораторная работа 5. Игровое поле «Дисковый лабиринт»

Лабораторные работы 6-8. Простой лабиринт. Динамический лабиринт

Лабораторная работа 9. Игровое поле «Перемещение фишек»

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.

Урок 10. Блок команд «Управление»

Урок 11. Проекты «Разрушение замка» и «Динамическое разрушение замка»

Урок 12. Проект «Детектор линии»

Лабораторная работа 10. Ветвления на базе платформы VEXcode VR

Лабораторная работа 11. Циклы на базе платформы VEXcode VR

Лабораторная работа 12. Блок «Всегда», блок «Прерывания» и блок «Ждать пока»

Лабораторные работы 13—15. Проект по уборке территории

Лабораторная работа 16. Поиск и подсчёт линий

Модуль 5. «Творческий проект»

При выполнении творческих проектных заданий школьники будут разрабатывать свои собственные программы.

Планы учебных занятий

Урок 2. Знакомство со средой VEXcode VR

Тип урока: комбинированный.

Цель урока: знакомство с интерфейсом VEXcode VR.

Планируемые результаты

Предметные: получение навыков работы с панелью управления, получение навыков по созданию первых программ в среде VEXcode VR.

Метапредметные: умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности; умение контролировать и корректировать учебную деятельность, способность ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их.

Личностные: готовность и способность обучающихся к саморазвитию и личностному самоопределению; сформированность их мотивации к обучению и целенаправленной познавательной деятельности.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в сеть Интернет, проектор, интерактивная доска.

Ход урока

1. Этап постановки цели и задач урока, мотивации к учебной деятельности — 15 минут.

Деятельность учителя — подача нового материала с демонстрацией платформы <http://vr.vex.com>.

VEXcode VR облегчает изучение информатики и робототехники, позволяя обучающимся кодировать виртуального робота, находясь в любом месте, программируя в среде блочного кодирования. VEXcode VR основан на VEXcode, том же интерфейсе программирования, который используется для робототехнических платформ VEX 123, GO, IQ и V5.

Показать удобство VEXcode VR для различных образовательных учреждений.

VEXcode VR — это способ обогатить опыт компьютерных наук для обучающихся после того, как они открыли для себя программирование в среде Scratch и азарт образовательной робототехники 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 обучающихся. Эти функции включают простоту использования, немедленную обратную связь, возможность сделать обучение видимым и помочь с различными реализациями в классе.

Легко использовать

VEXcode VR на 100% основан на веб-технологиях, поэтому запустить VEXcode VR просто. Пользовательский интерфейс (ПИ) упрощает навигацию — команды разделены на категории, ПИ построен, избегая большого списка команд в меню. Команды также имеют цветовую кодировку, поэтому пользователи могут легко находить связанные блоки. Область программирования всегда видна; приглашение учащихся начать программирование. VEXcode VR использует готовых роботов и команды трансмиссии. Это позволяет пользователям управлять своим виртуальным роботом за считанные секунды.

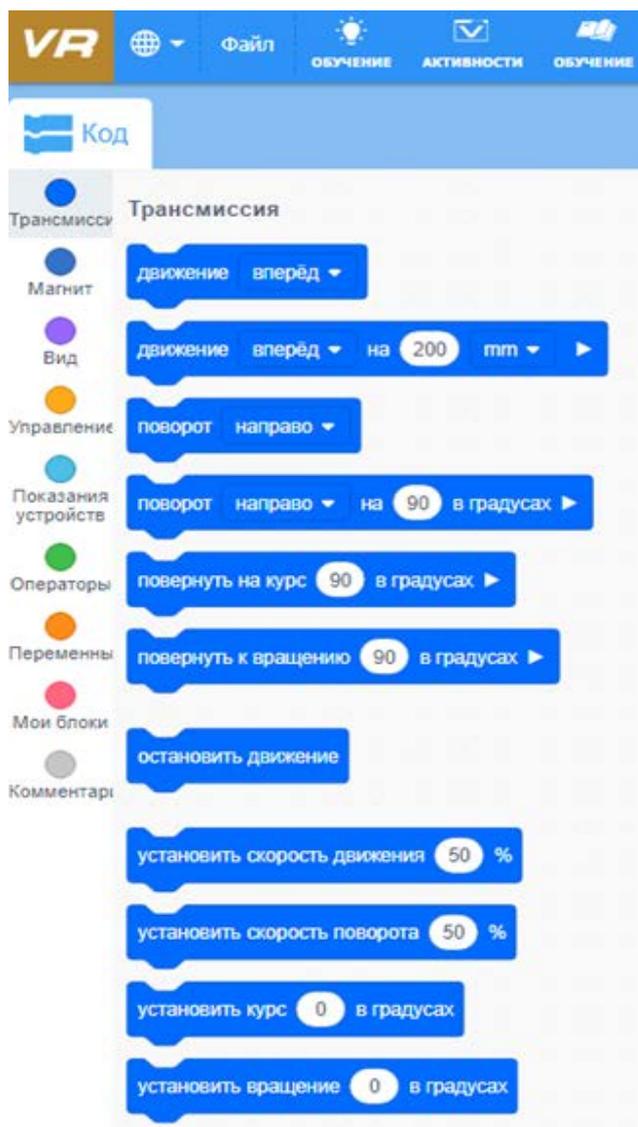


Рис. 3. Блок команд «Трансмиссия»

Робот VEX VR оснащён датчиками, элементами управления и множеством физических функций. В VEXcode VR есть только один робот, и он уже настроен заранее. Это устраняет необходимость в настройке конфигурации робота или заранее определённом шаблоне проекта.

Обратная связь

VEXcode VR стимулирует проведение экспериментов и игр. Когда обучающиеся запускают проект, они могут сразу увидеть, дал ли их робот желаемый результат. Учителя легко контролируют успехи обучающихся. Проект в VEXcode VR всегда будет работать одинаково — это не всегда происходит с физическим роботом. Это позволяет учителям и ученикам сосредоточиться на логике программирования, а не на физике робота или на игровом поле, на котором работает робот.

Игровое поле (площадка) VEXcode VR

Робот VR всегда начинает выполнение программы с одной и той же точки. Обучающиеся 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.

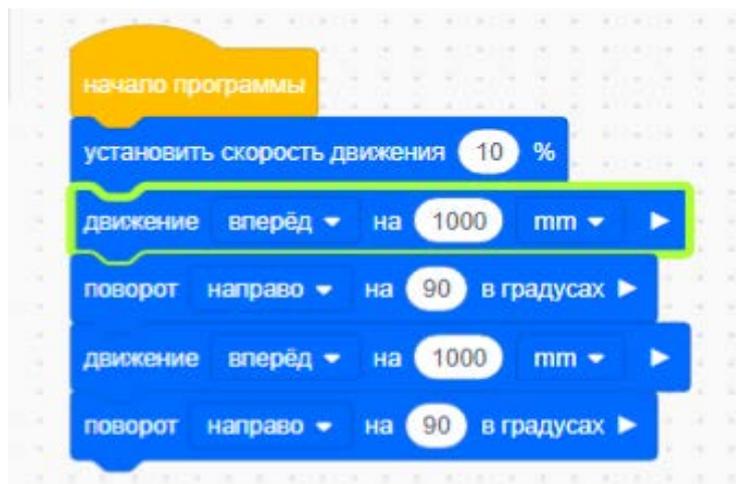


Рис. 4

В VEXcode VR ошибок нет. Обучающиеся могут делать логические ошибки при кодировании, но они не будут разочарованы тем, что их проекты не компилируются и не запускаются. Способность VEXcode VR обеспечивать немедленную обратную связь и простота использования побуждают обучающихся учиться в процессе написания кода создавать код с помощью небольших фрагментов, постепенно формируя окончательный вариант программы (скрипта).

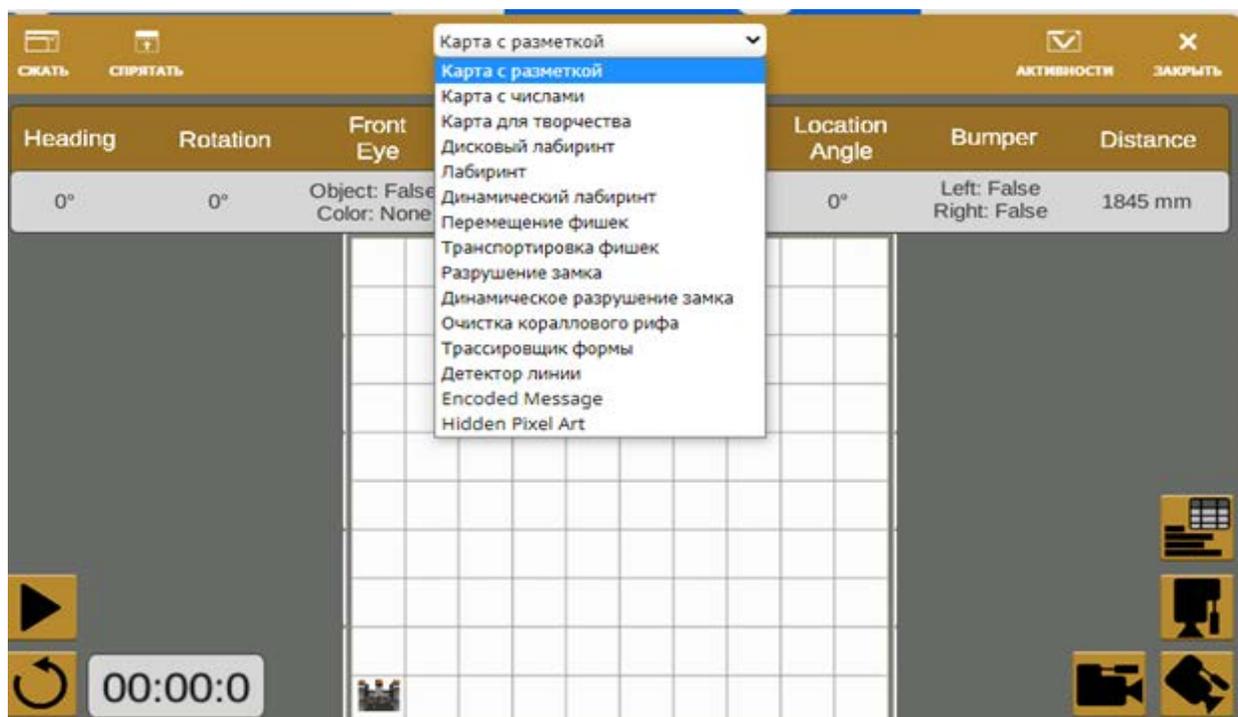


Рис. 5. Игровое поле с выпадающим меню выбора игрового поля

Обучение стало наглядным

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

РАСШИРИТЬ		СПРЯТАТЬ		Карта с разметкой				ВЫБЕРИТЕ ИГРОВОЕ ПОЛЕ		АКТИВНОСТИ		ЗАКРЫТЬ	
Heading	Rotation	Front Eye	Down Eye	Location		Location Angle	Bumper		Distance				
0°	0°	Object: False Color: None	Object: False Color: None	X: -900 mm	Y: -900 mm	0°	Left: False Right: False	1845 mm					

Рис. 6. Приборная панель с показаниями датчиков

VEXcode VR также выделяет блоки в проекте, когда эти блоки выполняются зелёным контуром. Эта функция позволяет учащимся наблюдать за ходом выполнения своих проектов. Когда проект VEXcode VR запущен, выполняемый блок окружён светящейся зелёной рамкой. Эта обратная связь помогает обучающимся понять, почему робот VR 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.

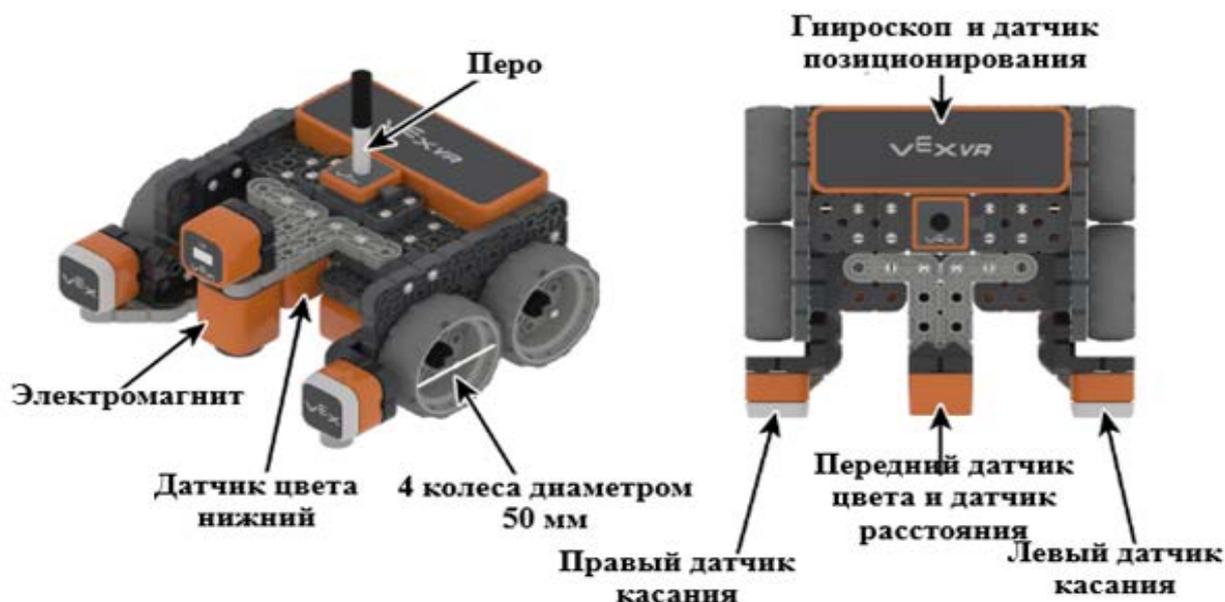


Рис. 7. Конфигурация датчиков робота

Управление роботом

Робот VR имеет следующие элементы управления:

- трансмиссия с гироскопом. Это включает категорию блоков «Трансмиссия» в наборе инструментов VEXcode VR;
- функция «рисования пером», позволяющая размещать перо вверх (чтобы не рисовать) или вниз (для рисования);
- электромагнит для поднятия дисков с металлическими сердечниками.

Физические характеристики робота

Робот VR обладает следующими физическими характеристиками:

- колёса имеют диаметр 50 мм;

- колёсная база (расстояние между центром переднего колеса и центром заднего колеса) составляет примерно 50,8 мм;
- Длина робота VR составляет 133 мм.

Датчики роботов

Робот VR оснащён следующими датчиками:

- моторные энкодеры с углом поворота 360° на оборот колеса;
- передний датчик глаза 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 робота VEXcode VR

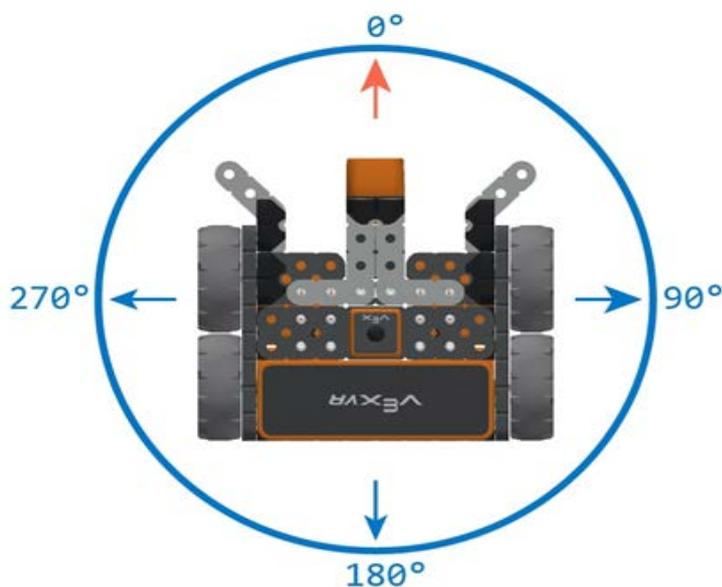


Рис. 8. Направления курса, задаваемые блоком «Курс»

Два датчика цвета, один обращён вперёд, а другой вниз. Эти датчики могут определять наличие объекта. Если есть, датчик также может определять цвет (красный, зелёный, синий, нет).

Датчик местоположения, который считывает координаты (X, Y) от центральной поворотной точки VR-робота.

2. Этап актуализации знаний и пробного учебного действия — 15 минут.

Учитель предлагает обучающимся под его контролем выполнить некоторые действия на платформе:

- запрограммировать робота на движение вперёд и назад. Блок программ «Трансмиссия»;
- запрограммировать робота на движение на определенное расстояние. Блок программ «Трансмиссия»;
- запрограммировать робота на повороты в градусах и по курсу. Блок программ «Трансмиссия»;
- проверить загрузку различных учебных полей.

3. Этап повторения нового материала — 5 минут.

Учитель повторяет кратко новый материал: роботы, датчики, скрипты, игровые поля, как заставить робота двигаться, поворачивать.

4. Этап проверки понимания и первичного закрепления — 5 минут.

Учитель задаёт вопросы ученикам:

Для чего нужна программа?

Что такое датчики и для чего они нужны?

5. Информация о домашнем задании, инструктаж по его выполнению — 3 минуты.

Задание 1. Зайти на платформу VEX code VR.

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

6. Этап рефлексии деятельности на уроке — 2 минуты.

Учитель интересуется у обучающихся их впечатлениями от урока, что им понравилось и что было непонятно.

Урок 3. Исполнительные механизмы конструкторов VEX

Тип урока: комбинированный.

Цель урока: создание простейших программ (скриптов), сохранение и загрузка проекта.

Планируемые результаты

Предметные: получение навыков написания программ (скриптов) на платформе VEXcode VR.

Метапредметные: умение соотносить свои действия с планируемыми результатами; осуществлять контроль своей деятельности в процессе достижения результата; определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией.

Личностные: стимулирование поиска вариантов на основе имеющихся знаний; готовность и способность обучающихся к саморазвитию и личностному самоопределению; развитие находчивости, умения преодолевать трудности для достижения намеченной цели.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в сеть Интернет, проектор, интерактивная доска.

Ход урока

1. Этап постановки цели и задач урока, мотивации к учебной деятельности 15 — минут.

Деятельность учителя — подача нового материала с демонстрацией платформы <http://vr.vex.com>.

Примеры различных программ, созданных с помощью платформы, можно открыть в меню **Файл-Открыть примеры**.

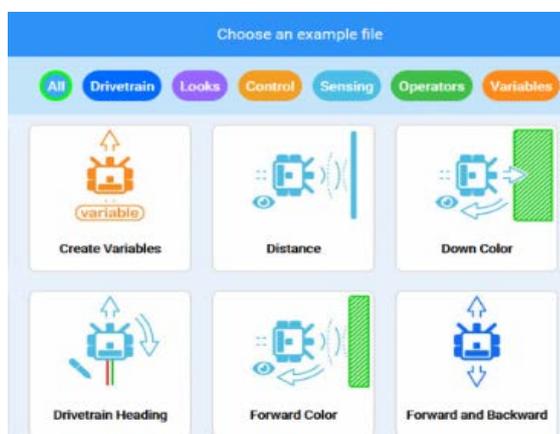


Рис. 9. Примеры программ, созданных на платформе VEXcode VR

Создайте простую программу движения робота по периметру игрового поля **Карта с разметкой** (размеры клетки 200×200 мм).

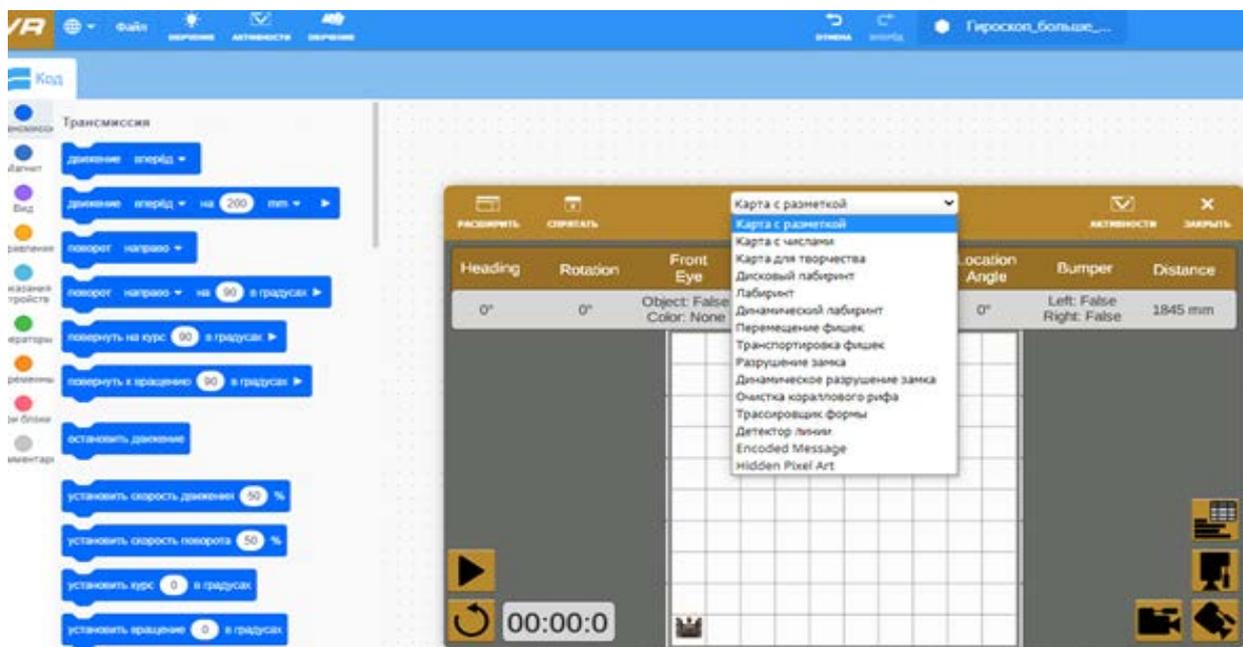


Рис. 10. Игровое поле **Карта с разметкой**

Простая программа будет выглядеть следующим образом:

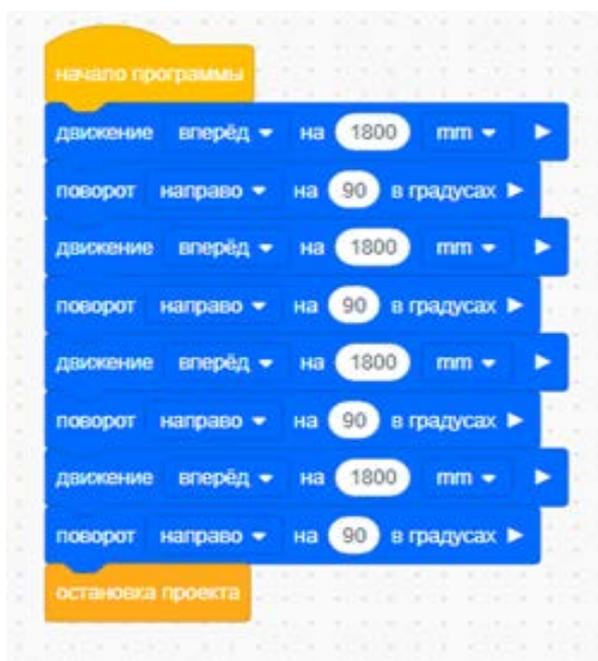


Рис. 11. Скрипт обхода игрового поля по периметру

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

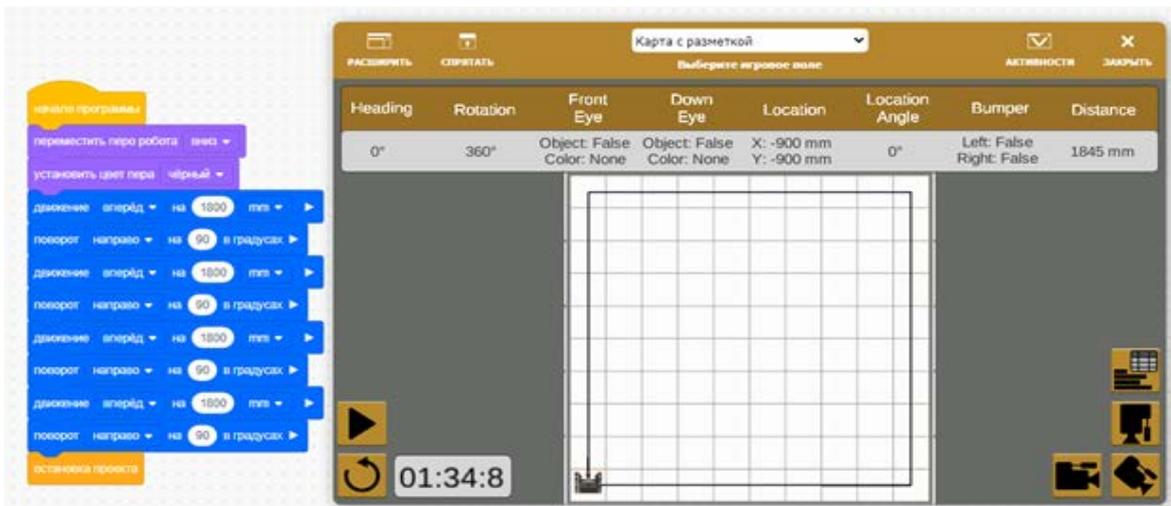


Рис. 12. Скрипт рисования роботом квадрата

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

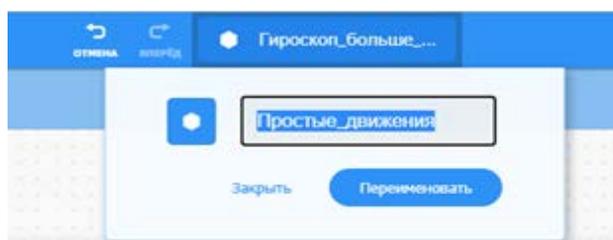


Рис. 13. Переименование рабочего файла

Сохранить этот файл на своё устройство можно через меню **Файл-Сохранить на ваше устройство**, файл будет скачан в папку **Загрузки** рабочего устройства.

Также рассмотрим позиционирование робота на игровой площадке с помощью датчика положения (гироскопа).

Игровая площадка разделена на координатные плоскости (декартовы координаты) как показано на рисунке 14.

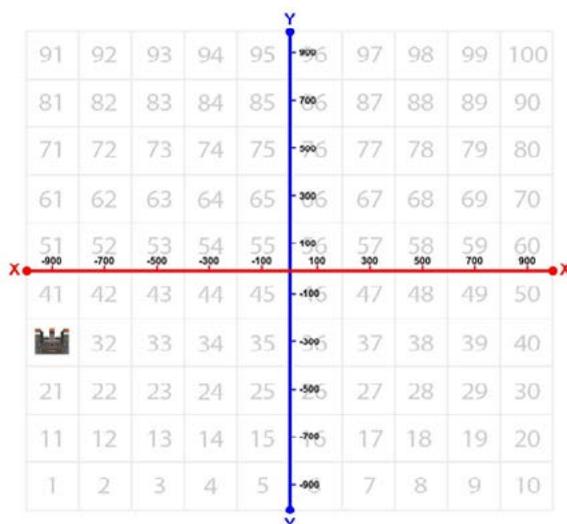


Рис. 14. Координатная плоскость игрового поля

Движение по координатной плоскости задаётся следующим образом:

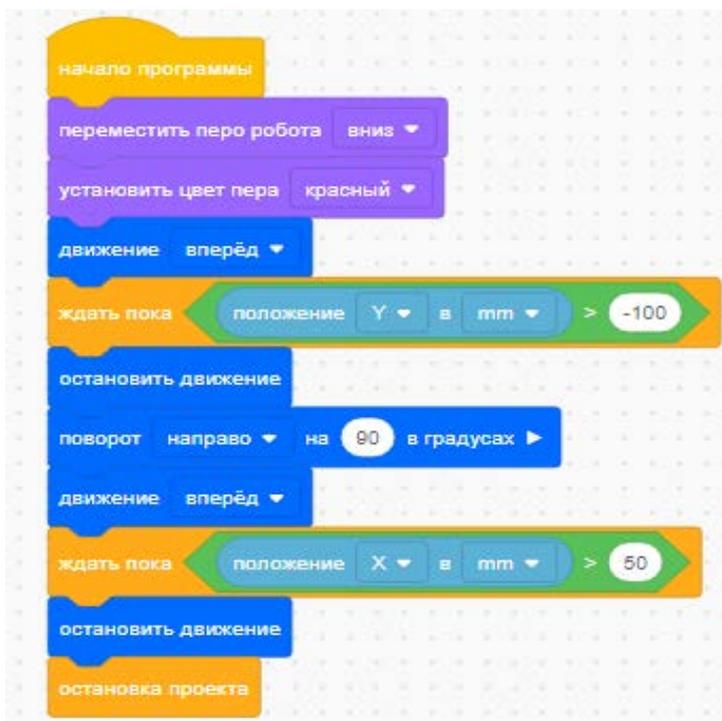


Рис. 15. Перемещение робота по координатам

2. Этап актуализации знаний и пробного учебного действия — 15 минут.

Учитель предлагает обучающимся под его контролем выполнить некоторые действия на платформе, используя знания, полученные в ходе изложения нового материала.

3. Этап актуализации знаний и пробного учебного действия — 15 минут.

Учитель предлагает обучающимся под его контролем выполнить некоторые действия на платформе:

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

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

4. Этап повторения нового материала — 5 минут.

Учитель повторяет кратко новый материал: программирование простых движений робота, датчики, скрипты, игровые поля, как заставить робота двигаться, поворачивать, рисовать.

5. Этап проверки понимания и первичного закрепления — 5 минут.

Учитель задаёт вопросы ученикам:

- 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?
- Как управлять магнитом и для чего?

6. Информация о домашнем задании, инструктаж по его выполнению — 3 минут.

Задание. Написать программу рисования на игровом поле роботом инициалов обучающегося.

7. Этап рефлексии деятельности на уроке — 2 минут.

Учитель интересуется у обучающихся об их впечатлениях от урока, что им понравилось и что было непонятно.

Урок 10. Блок команд «Управление»

Цель урока: ознакомить обучающихся с ветвлениями и циклами.

Тип урока: комбинированный.

Планируемые результаты

Предметные: получение навыков по использованию условного оператора и операторов цикла, разработка программ, реализующих разветвляющийся и циклический алгоритмы.

Метапредметные: делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи.

Личностные: формирование мотивации к обучению и целенаправленной познавательной деятельности; формирование целеустремлённости и усидчивости в процессе творческой, исследовательской работы и учебной деятельности.

Время реализации: 1 академический час.

Оборудование и материалы: компьютеры с выходом в сеть Интернет, проектор, интерактивная доска

Ход урока

2. Этап постановки цели и задач урока, мотивации к учебной деятельности — 15 минут.

Деятельность учителя: подача нового материала с демонстрацией командам управления в среде vr.vex.com.

Блок команд управления в VEX VR

Вы уже знакомы с понятиями циклов и ветвлений, поэтому самое время перейти к изучению этих структур в среде программирования VEX 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 нельзя получить интеллектуальную систему, способную ориентироваться в пространстве без дополнительной помощи со стороны оператора или команды разработчиков. На рисунке 16 показан блок команд управления. Обратите внимание, что многие блоки схожи между собой по названиям и, соответственно, обладают похожими свойствами. Сегодня познакомимся с блоками **Ждать** и **Повторять** (рис. 16).

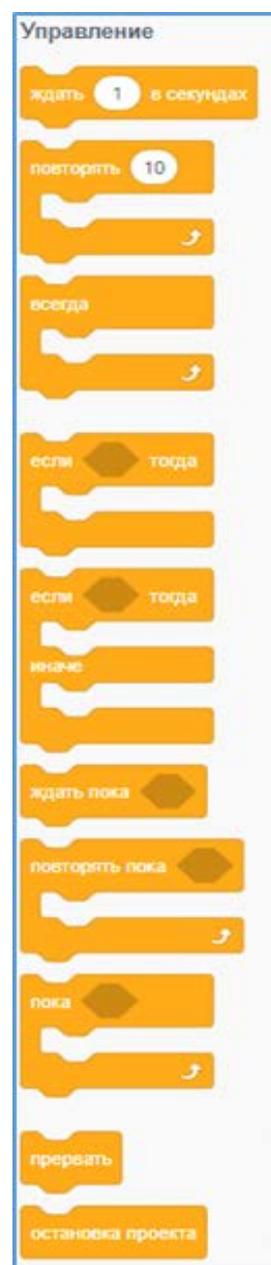


Рис. 16. Блок команд управления

Рассмотрим блок **Ожидания** (рис. 17).



Рис. 17. Блок **Ожидания**

Данный блок позволяет удерживать действия, которые происходили до его вызова. Рассмотрим пример (рис. 18).

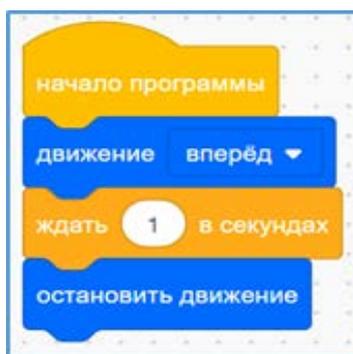


Рис. 18. Пример программы с использованием блока **Ожидания**

Что произойдёт в данном отрывке кода? Робот будет двигаться вперед, после чего остановится. Если же убрать данный блок, то, соответственно, 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 к блоку **Повторять**. На самом деле, как вы уже поняли, данный блок повторяет некоторое число действий, количество которых определено внутри этого блока. Рассмотрим программу для изображения квадрата (рис. 19).

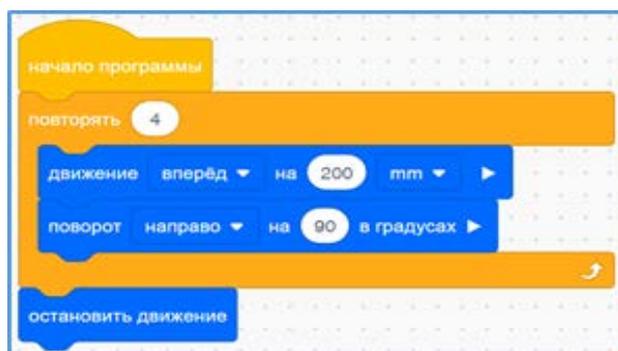


Рис. 19. Программа по объезду квадрата

Давайте более подробно рассмотрим данную программу. Почему бы просто не использовать группу из одинаковых блоков для решения задачи? Никто не запрещает использовать несколько одинаковых блоков вместо циклов, но при написании кода большего объёма получим гораздо больше строк, что, несомненно, усложнит читаемость всей программы. Для сравнения на рисунке 20 показан тот же самый код, но без использования цикла.

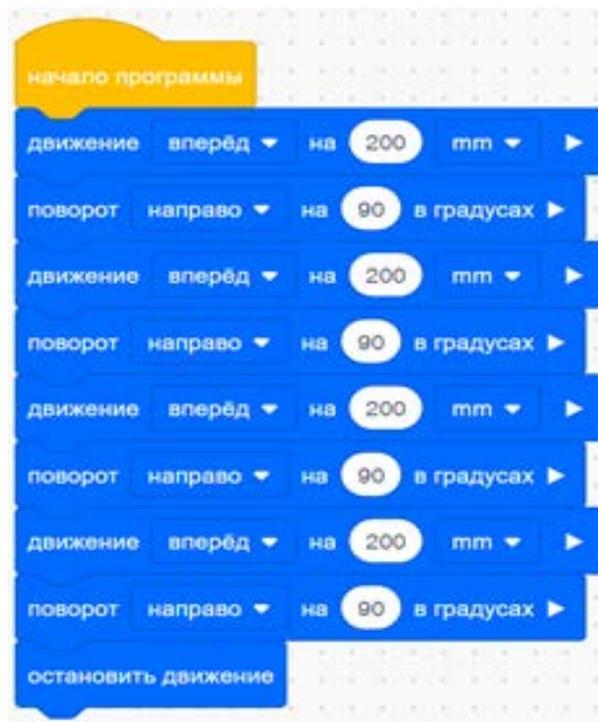


Рис. 20. Объезд по квадрату без цикла

Как вы видите, циклы упрощают читаемость кода, не влияя на его функциональность.

3. Этап актуализации знаний и пробного учебного действия — 15 минут.

Учитель предлагает обучающимся под его контролем выполнить некоторые действия на платформе:

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

4. Этап повторения нового материала — 5 минут.

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

5. Этап проверки понимания и первичного закрепления — 5 минут.

Учитель задаёт вопросы ученикам:

- Почему блоки управления так называются?
- Зачем нужны блоки управления?
- Что позволяет делать блок ожидания?
- Что делает блок повторения?

6. Информация о домашнем задании, инструктаж по его выполнению — 3 минут.

Задание 1. Реализовать алгоритм объезда роботом треугольника и прямоугольника.

Задание 2. Реализовать алгоритм объезда роботом треугольника и прямоугольника без использования блоков поворота с указанием угла поворота (необходимо использовать только блок ожидания).

7. Этап рефлексии деятельности на уроке — 2 минуты.

Учитель интересуется у обучающихся об их впечатлениях от урока, что им понравилось и что было непонятно.

Лабораторная работа 10 Ветвления на базе платформы VEXcode VR

Теоретическая часть

Иногда роботу бывает необходимо единожды принять выбор относительно своего текущего действия. Это могут быть повороты направо или налево, отъезды вперёд или назад в определённый момент времени без необходимости перепроверки действий, как в циклах. В этом случае лучше всего использовать блоки **Если-тогда** и **Если-тогда-иначе**, которые позволяют преобразовать действия, совершаемые роботом, в систему с выходами да/нет.

Практическая часть

Цель работы: ознакомить обучающихся с ветвлениями на базе платформы vr.vex.com.

Ход лабораторной работы

На предыдущем занятии рассмотрели блоки ожидания и повторения и сегодня продолжим изучение остальных блоков модуля управления.

В первую очередь обратим внимание на блок **Если-тогда** (рис. 21).



Рис. 21. Блок Если-тогда

Тело данного блока выполняется в том случае, если условие, обозначенное в шестиугольной вставке, является правдой, и не выполняется в обратном случае. Рассмотрим следующий пример (рис. 22).



Рис. 22. Пример использования условного блока

Давайте подробнее рассмотрим, что будет происходить в данном примере. Как вы видите, в качестве условия в блоке дано логическое выражение, содержащее справа сумму чисел, а слева константу. Если сравнение окажется правдой, то в окно вывода будет напечатано True, если условие окажется ложным — False.

Кроме того, существует ещё один блок управления, схожий с вышеуказанным, но при этом имеющий ещё одну дополнительную ветвь иначе (рис. 23).



Рис. 23. Блок **Если-тогда-иначе**

Рассмотрим следующую программу для понимания разницы между блоками **Если-тогда** и **Если-тогда-иначе** (рис. 24).

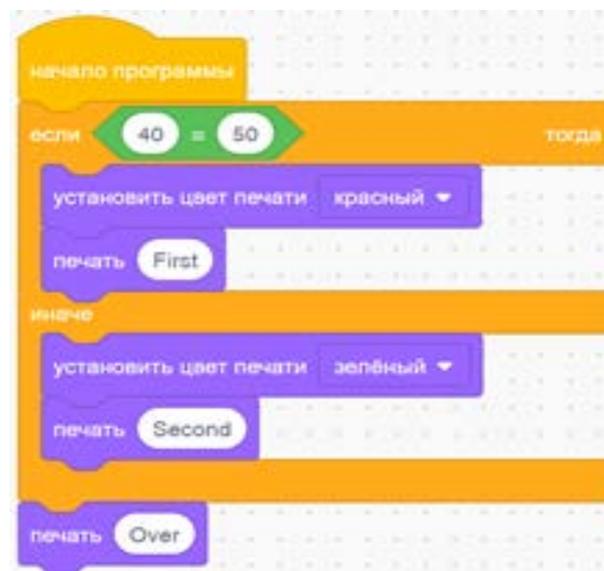


Рис. 24. Демонстрация работы **Если-тогда-иначе**

Рассмотрим код данной программы. Выражение *после если* в данном случае является ложным, поэтому будет выполняться часть после *иначе*, то есть перо робота станет зелёного цвета и на экране появится надпись Second. Обратите внимание, что Over будет выведено в любом случае, так как находится за пределами блока условного оператора.

Задание 1. Написать программу для вывода на экран True или False в зависимости от решения простых математических выражений, которые учащиеся должны разместить внутри условия условного блока.

Задание 2. Реализовать программу для поворота роботом перед препятствием на произвольный угол.

Контрольные вопросы:

1. Как можно использовать блок **Если-тогда**?
2. В чём отличие блоков **Если-тогда** и **Если-тогда-иначе**?
3. В каких ситуациях лучше всего использовать блок **Если-тогда-иначе**?

Лабораторная работа 11 Циклы на базе платформы VEXcode VR

Теоретическая часть

Известно, что бывают ситуации, при которых необходимо проверять условие несколько раз, прежде чем принять окончательное решение. Можно использовать совокупность блоков повторения и условные блоки, но для корректной работы повторений должно быть огромное число (десятки миллионов) и подбор точного числа может сильно затруднить общее время создания программы. Во избежание этого в VEX VR используются циклы **Повторять пока** и **Пока**.

Практическая часть

Цель работы: ознакомить обучающихся с циклами на базе платформы vr.vex.com

Ход лабораторной работы

В VEX VR используются циклы **Повторять пока** и **Пока** (рис. 25).

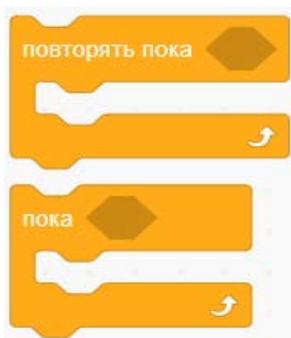


Рис. 25 — Блоки циклов

Рассмотрим простой пример программы, который покажет отличия между двумя видами циклов (рис. 26).



Рис. 26. Отличия циклов в VEX VR

На первый взгляд может показаться, что отличия в условиях блоков циклов должны привести к выполнению абсолютно разных действий, но это не так. Природа блока **Повторять пока** такова, что действия внутри блока будут выполняться до тех пор, пока условие будет ложным. Об этом важно помнить при составлении программы.

Давайте пока отложим блок **Повторять пока** и напишем простую программу для вывода на экран нескольких строк текста в зависимости от показаний датчика расстояния (рис. 27).

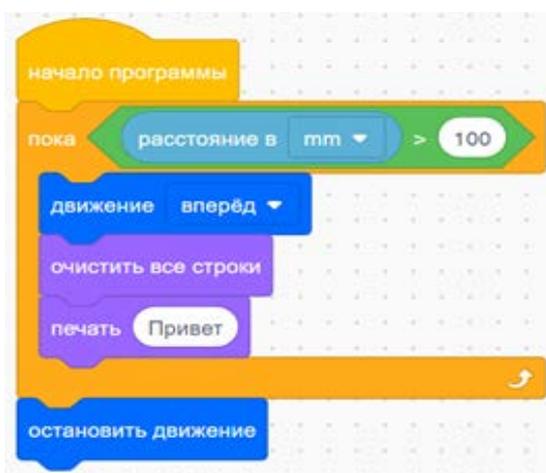


Рис. 27. Работа с циклом Пока

Давайте построчно рассмотрим данную программу. В первую очередь создаём 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» огромное число раз в силу высокой производительности управляющего контроллера. Если добавить к программе всего одну строку, то слово «Привет» будет выведено только один раз (рис. 28).

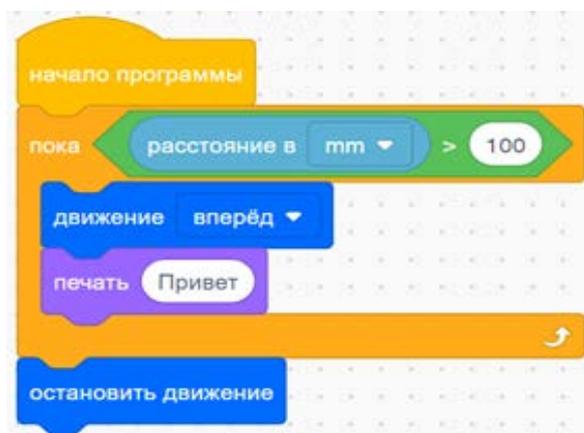


Рис. 28. Вывод одного слова на экран во время использования цикла

Задание 1. Написать программу для вывода на экран *привет*. При каждом проходе цикла привет должно выводиться с новой строки.

Задание 2. Написать программу по изображению роботом квадрата (в качестве повторения пройденного материала).

Контрольные вопросы:

1. В чём отличия блоков **пока** и **повторять пока**?
2. Приведите примеры программы, использующей блок **пока** (в виде псевдокода).
3. Приведите примеры программы, использующей блок **повторять пока** (в виде псевдокода).

Лабораторная работа 12

Блок Всегда, блок Прерывания и блок Ждать пока

Теоретическая часть

На предыдущем занятии рассмотрели блоки циклов и сегодня продолжим изучение оставшихся блоков модуля управления: блока **Всегда**, блока **Прерывания** и блока **Ждать пока**.

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

Практическая часть

Цель работы: ознакомить обучающихся с блоками **Всегда**, блоками **Прерывания** и блоками **Ждать пока**.

Ход лабораторной работы

Рассмотрим две программы, которые являются абсолютно идентичными, но при этом продемонстрируют принцип работы блока **Всегда** (рис. 29).

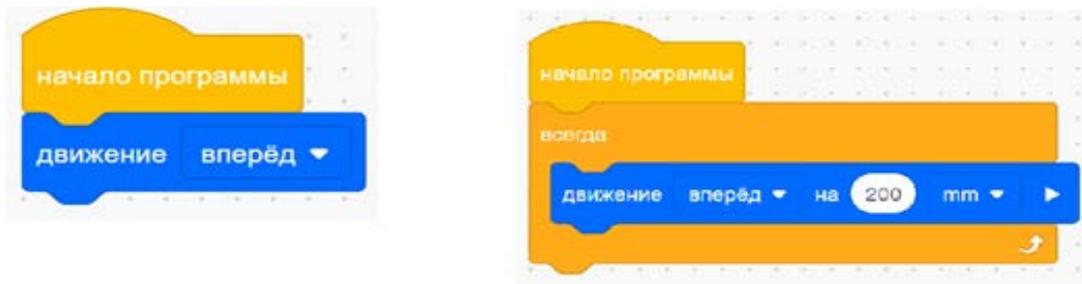


Рис. 29. Аналог бесконечного цикла

Как вы можете заметить, в данной программе блок движения включает себя бесконечный цикл, что позволяет роботу двигаться бесконечно вперед при добавлении этого блока в программу.

Для того чтобы прервать выполнение бесконечного цикла, можно использовать блок **Прерывания**.

Рассмотрим программу, в которой робот будет ехать вперед до тех пор, пока датчик расстояния не увидит перед собой некоторое препятствие на расстоянии 100 мм (рис. 30).



Рис. 30. Программа для демонстрации работы цикла прерывания

Данная программа выполняет все те же действия, что и использование цикла Пока в предыдущем уроке, но при этом наглядно демонстрирует способ прерывания цикла Всегда. Использование блока **Прервать** помогает избежать поломок робота, так как обычно используется в контексте аварийного прерывания выполнения программы. Давайте рассмотрим более сложный вариант программы, в котором использование цикла будет оправдано срабатыванием датчика расстояния на расстоянии меньшем, чем планируемое (рис. 31).

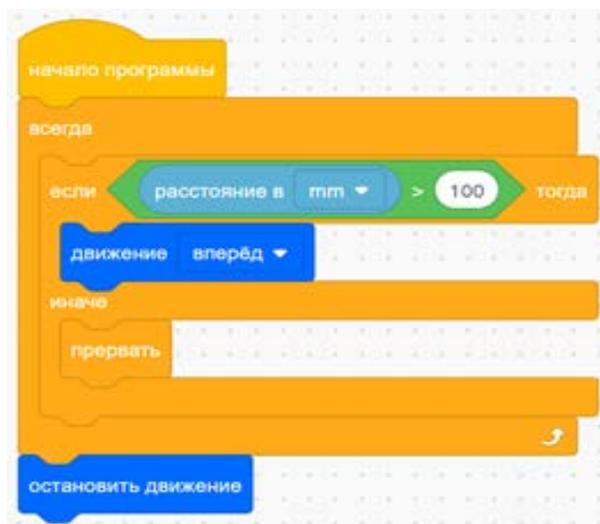


Рис. 31. Пример программы

В данной программе робот будет совершать поворот до тех пор, пока не обнаружит перед собой препятствие. После этого он отъедет назад на 200 мм и остановит своё движение. Данный алгоритм может быть применим к случаям, когда на роботе установлено техническое зрение и ему необходим лучший обзор для идентификации перед собой объектов.

Кроме того, необходимо обратить внимание на блок **Ждать пока** (рис. 32), который позволяет удержать выполнение действий до этого блока.

Рассмотрим программу, которая может помочь в прохождении некоторых испытаний в среде VEX VR.

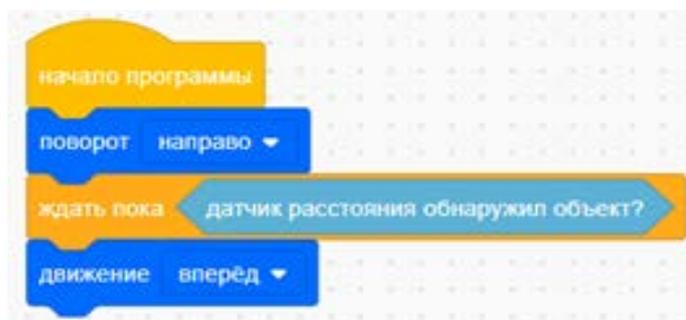


Рис. 32. Блок **Ждать пока**

В данной программе робот при обнаружении объекта постарается вытолкнуть его с поля. Данный код поможет в решении задач на последующих уроках.

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

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

Задание 3. Реализовать программу для активации роботом отъезда назад при первоначальном движении робота вперёд.

Контрольные вопросы:

1. Зачем нужен блок **Прерывания**?
2. Приведите примеры программы, использующей блок **Прерывания** (в виде псевдокода)
3. Как работает блок **Ждать пока**?

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).
4. STEM Education channel by Mark Johnston // URL: <https://www.mjstem.com/> (Дата обращения: 15.04.2021).

Перечень доступных источников информации

1. <https://scratch.mit.edu/> Сообщество Sctach.
2. Python для начинающих 2021 – уроки, задачи и тесты <https://pythonru.com/uroki/python-dlja-nachinajushhih>
3. Python/Учебник Python 3.1 [https://ru.wikibooks.org/wiki/ Python/%D0%A3%D1%87%D0%B5%D0%B1%D0%BD%D0%B8%D0%BA_Python_3.1](https://ru.wikibooks.org/wiki/Python/%D0%A3%D1%87%D0%B5%D0%B1%D0%BD%D0%B8%D0%BA_Python_3.1)
4. Босова Л. Л. Информатика. 8 класс: учебник. – М.: БИНОМ. Лаборатория знаний, 2016. – 176 с.
5. Буйначев С. К. Основы программирования на языке Python: учеб. пособие. – Екатеринбург: Изд-во Урал. ун-та, 2014. – 91 с.
6. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. – СПб.: Питер, 2017. – 288 с.
7. Бэрри П. Изучаем программирование на Python. – М., 2017. – 624 с.
8. Винницкий Ю. А. Scratch и Arduino для юных программистов и конструкторов. – СПб: БХВ-Петербург, 2018. – 176 с.
9. Голиков Д. В. Scratch для юных программистов. – СПб.: БХВ-Петербург, 2017. – 192 с.
10. Гэддис Т. Начинаем программировать на Python / Пер. с англ. – 4-е изд. – СПб.: БХВ-Петербург, 2019. – 768 с.
11. Лаборатория юного линуксоида. Введение в Scratch. <http://younglinux.info/scratch>
12. Луридас П. Алгоритмы для начинающих: теория и практика для разработчика. – М. : Эксмо, 2018. – 608 с.
13. Лутц М. Изучаем Python / Пер. с англ. – 3-е изд – СПб.: Символ Плюс, 2009. – 848 с.
14. Маржи М. Scratch для детей. Самоучитель по программированию – пер. с англ. М. Гескиной и С. Таскаевой. – М. : Манн, Иванов и Фербер, 2017. – 288 с.
15. Мюллер Дж. Python для чайников. – СПб.: Диалектика, 2019. – 416 с.
16. Пашковская Ю. В. Творческие задания в среде Scratch. Рабочая тетрадь для 5–6 классов. – М., 2018. – 195 с.
17. Первин Ю. А. Методика раннего обучения информатике. – М.: «Бином», Лаборатория базовых знаний, 2008. – 228 с.
18. Поляков К. Ю. Информатика. 7 класс (в 2 частях) : учебник. Ч. 1 / К. Ю. Поляков, Е. А. Еремин. – М.: БИНОМ. Лаборатория знаний, 2019. – 160 с.
19. Практический Python 3 для начинающих <https://pythonworld.ru/samouchitel-python>.
20. Рафгарден Т. Совершенный алгоритм. Жадные алгоритмы и динамическое программирование. – СПб.: Питер, 2020. – 256 с.
21. Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 336 с.
22. Рындак В. Г., Дженжер В. О., Денисова Л. В. Проектная деятельность школьника в среде программирования Scratch: учебно-метод. пособие. – Оренбург: Оренб. гос. ин-т менеджмента, 2009. – 116 с.
23. Свейгарт Эл. Программирование для детей. Делай игры и учи язык Scratch!. – М.: Эксмо, 2017. – 304 с.
24. Семакин И. Г., Залогова, Л. А. и др. Информатика и ИКТ: учебник для 9 класса. – М.: Бином, 2014. – 171 с.
25. Торгашева Ю. Первая книга юного программиста. Учимся писать программы на Scratch. – СПб.: Питер, 2016. – 128 с.

26. *Уфимцева П. Е., Рожина И. В.* Обучение программированию младших школьников в системе дополнительного образования с использованием среды разработки Scratch // Наука и перспективы. — 2018. — № 1. — с. 29—35.
27. Учебник по языку программирования Python (хабраиндекс) <https://habr.com/ru/post/61905/>
28. *Федоров Д. Ю.* Программирование на языке высокого уровня Python: учеб. пособие для прикладного бакалавриата. — М. : Издательство Юрайт, 2019. — 161 с.
29. *Адаменко А. Н., Кучуков А. М.* Логическое программирование и Visual Prolog. — СПб.: БХВ-Петербург, 2003. — 992 с.
30. *Братко И.* Программирование на языке Visual Prolog для искусственного интеллекта. — М.: Мир, 1990. — 560 с.
31. *Ин Ц., Соломон Д.* Использование Турбо—Пролог. — М.: Мир, 1993. — 608 с.
32. *Стерлинг Л., Шапиро Э.* Искусство программирования на языке Visual Prolog. — М.: Мир, 1990. — 235 с.