

Обучение программированию детей 6+



A decorative graphic on the left side of the slide consists of several thick, curved arrows in various colors (purple, red, blue, yellow, green) pointing upwards and to the right, creating a sense of movement and progress.

Обучение программированию детей 6–10 лет

1. Понятия «программа», «алгоритм», виды алгоритмов.
2. Создание алгоритмов в программах LEGO Education WeDo и RobboScratch.
3. Примеры заданий для детей 6+ по обучению программированию.
4. Программирование роботов Robbo.
5. Экспериментирование на занятиях по робототехнике.

Программирование

Программирование – это процесс создания компьютерных программ.

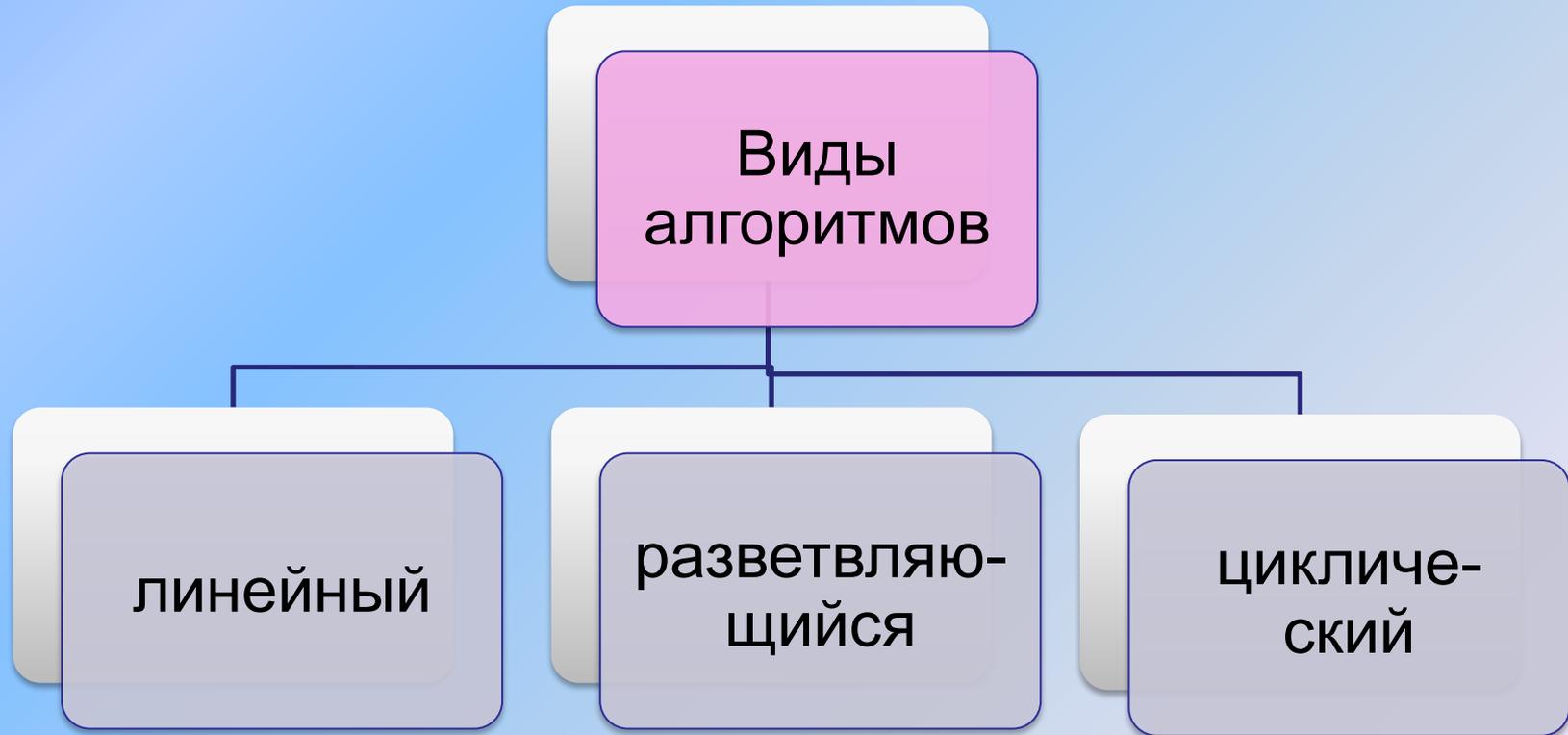
Программа, по мнению швейцарского учёного Никлауса Вирта, = алгоритмы + структуры данных.

Программирование основывается на использовании искусственных языков, на которых записываются исходные текста программ.

Программа – алгоритм, записанный на «понятном» компьютеру языке программирования.

Создание алгоритмов

Алгоритм – описание последовательности действий, строгое исполнение которых приводит к решению поставленной задачи за конечное число шагов.



Программирование

Линейный алгоритм – описание действий, которые выполняются однократно в заданном порядке.

Циклический алгоритм – описание действий, которые должны повторяться указанное число раз или пока не выполнено задание.

Разветвляющийся (ветвящийся) алгоритм – алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий

Объект, который будет выполнять алгоритм, обычно называют исполнителем.

Программирование

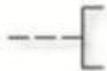


Формы представления алгоритмов:

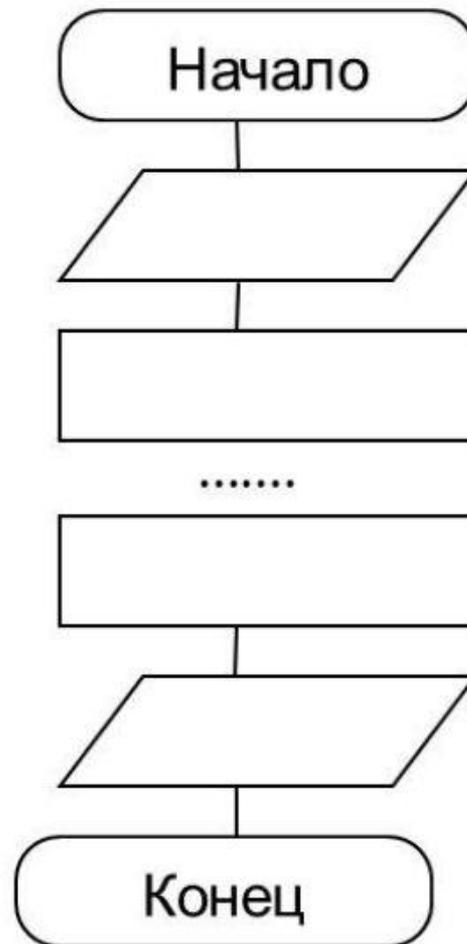
- ✓ словесные (запись на естественном языке);
- ✓ графические (изображения из графических символов);
- ✓ псевдокоды (полуформализованные описания алгоритмов на условном алгоритмическом языке, включающие в себя как элементы языка программирования, так и фразы естественного языка, общепринятые математические обозначения и др.);
- ✓ программные (тексты на языках программирования).

Графический способ представления алгоритмов является более компактным и наглядным по сравн. со словесным.

При графическом способе алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного или нескольких действий (схемой или блок-схемой).

Наименование	Обозначение	Функция
Блок начало-конец (пуск-остановка)		Элемент отображает выход во внешнюю среду и вход из внешней среды (наиболее частое применение – начало и конец программы). Внутри фигуры записывается соответствующее действие.
Блок действия		Выполнение одной или нескольких операций, обработка данных любого вида (изменение значения данных, формы представления, расположения). Внутри фигуры записывают непосредственно сами операции, например, операцию присваивания: $a = 10 * b + c$.
Логический блок (блок условия)		Отображает решение или функцию переключательного типа с одним входом и двумя или более альтернативными выходами, из которых только один может быть выбран после вычисления условий, определенных внутри этого элемента. Вход в элемент обозначается линией, входящей обычно в верхнюю вершину элемента. Если выходов два или три, то обычно каждый выход обозначается линией, выходящей из оставшихся вершин (боковых и нижней). Если выходов больше трех, то их следует показывать одной линией, выходящей из вершины (чаще нижней) элемента, которая затем разветвляется. Соответствующие результаты вычислений могут записываться рядом с линиями, отображающими эти пути. Примеры решения: в общем случае – сравнение (три выхода: $>$, $<$, $=$); в программировании – условные операторы <code>if</code> (два выхода: <code>true</code> , <code>false</code>) и <code>case</code> (множество выходов).
Предопределённый процесс		Символ отображает выполнение процесса, состоящего из одной или нескольких операций, который определен в другом месте программы (в подпрограмме, модуле). Внутри символа записывается название процесса и передаваемые в него данные. Например, в программировании – вызов процедуры или функции.
Данные (ввод-вывод)		Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод). Данный символ не определяет носителя данных (для указания типа носителя данных используются специфические символы).
Граница цикла		Символ состоит из двух частей – соответственно, начало и конец цикла – операции, выполняемые внутри цикла, размещаются между ними. Условия цикла и приращения записываются внутри символа начала или конца цикла – в зависимости от типа организации цикла. Часто для изображения на блок-схеме цикла вместо данного символа используют символ условия, указывая в нём решение, а одну из линий выхода замыкают выше в блок-схеме (перед операциями цикла).
Соединитель		Символ отображает вход в часть схемы и выход из другой части этой схемы. Используется для обрыва линии и продолжения её в другом месте (для избежания излишних пересечений или слишком длинных линий, а также, если схема состоит из нескольких страниц). Соответствующие соединительные символы должны иметь одинаковое (при том уникальное) обозначение.
Комментарий		Используется для более подробного описания шага, процесса или группы процессов. Описание помещается со стороны квадратной скобки и охватывается ей по всей высоте. Пунктирная линия идет к описываемому элементу, либо группе элементов (при этом группа выделяется замкнутой пунктирной линией). Также символ комментария следует использовать в тех случаях, когда объём текста, помещаемого внутри некоего символа (например, символ процесса, символ данных и др.), превышает размер самого этого символа.

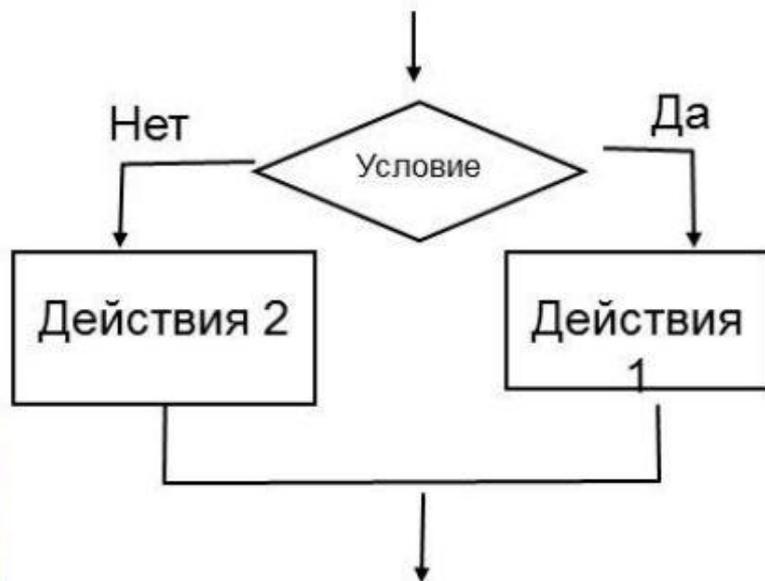
Блок-схема линейного алгоритма



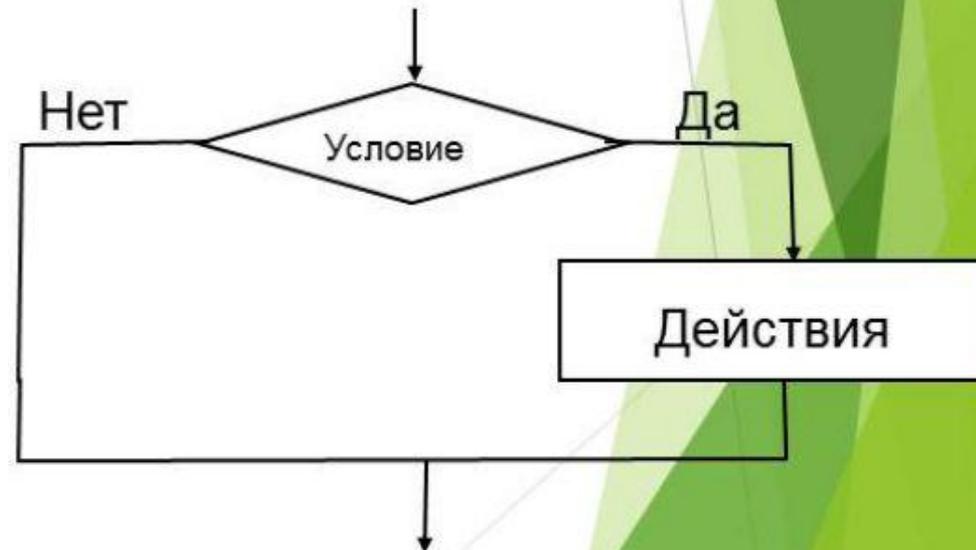
Блок - схема разветвляющегося алгоритма



Полное ветвление позволяет организовать в алгоритме две ветви (ТО или ИНАЧЕ)



Неполное ветвление предполагает наличие действий только на одной ветви (ТО), вторая ветвь отсутствует.



Блок-схема циклического алгоритма

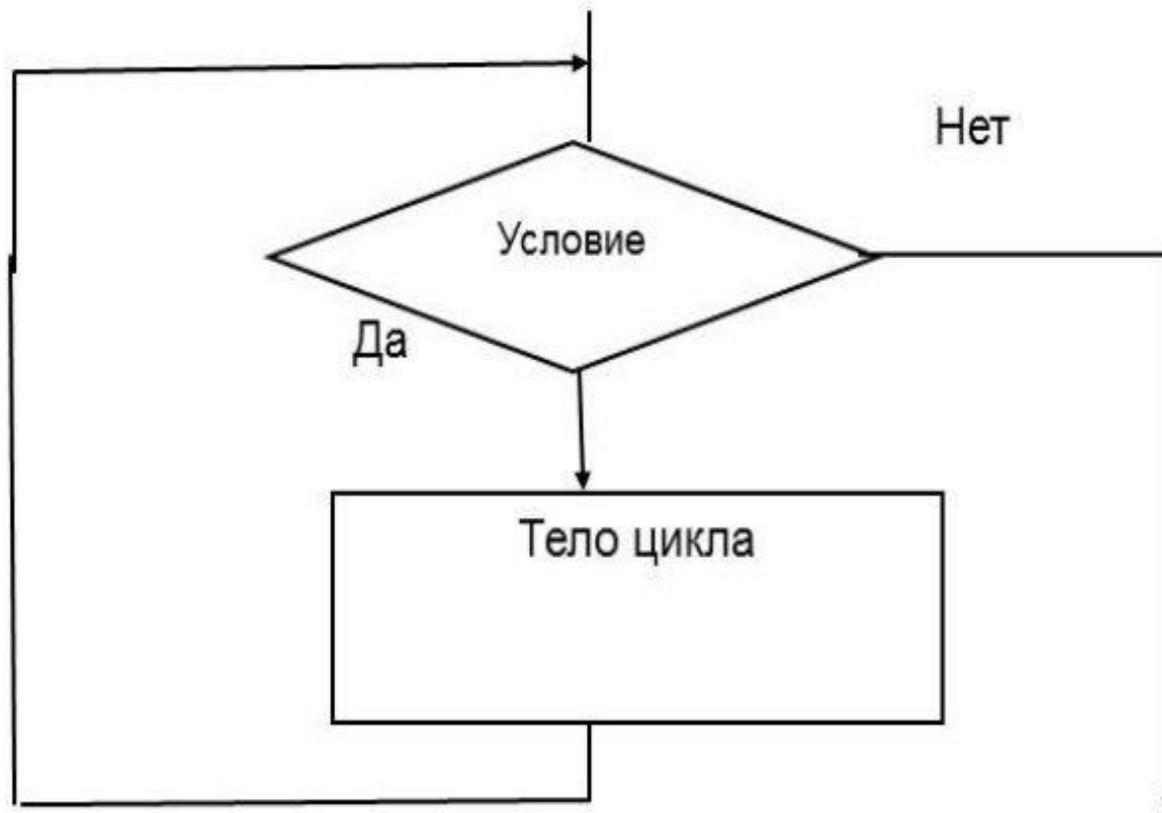


В цикле с параметром число повторений цикла однозначно определено и задается с помощью начального, конечного значений параметра и шагом его изменения.



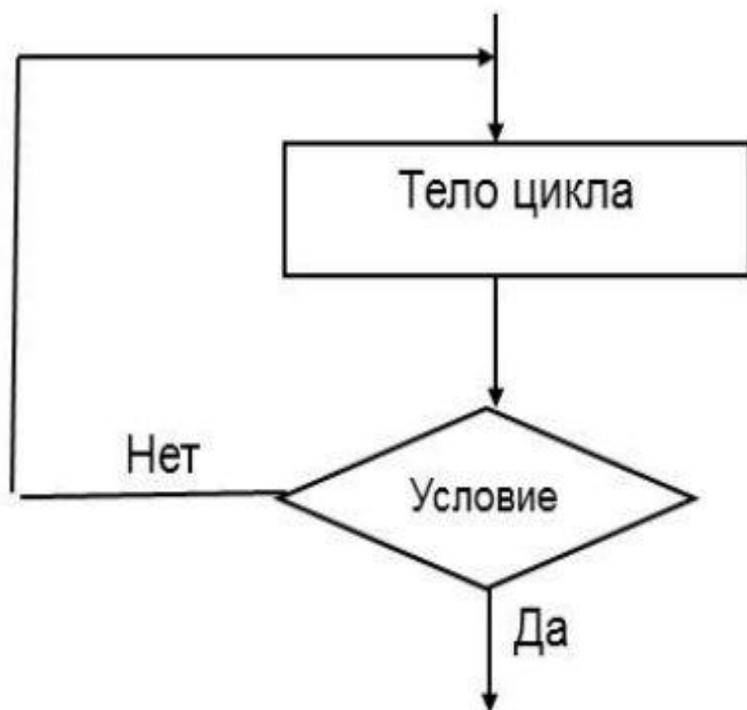
Цикл с предусловием

Действия внутри этого цикла повторяются, пока выполняется условие в блоке ветвления, причем сначала проверяется условие, а затем выполняется действие.



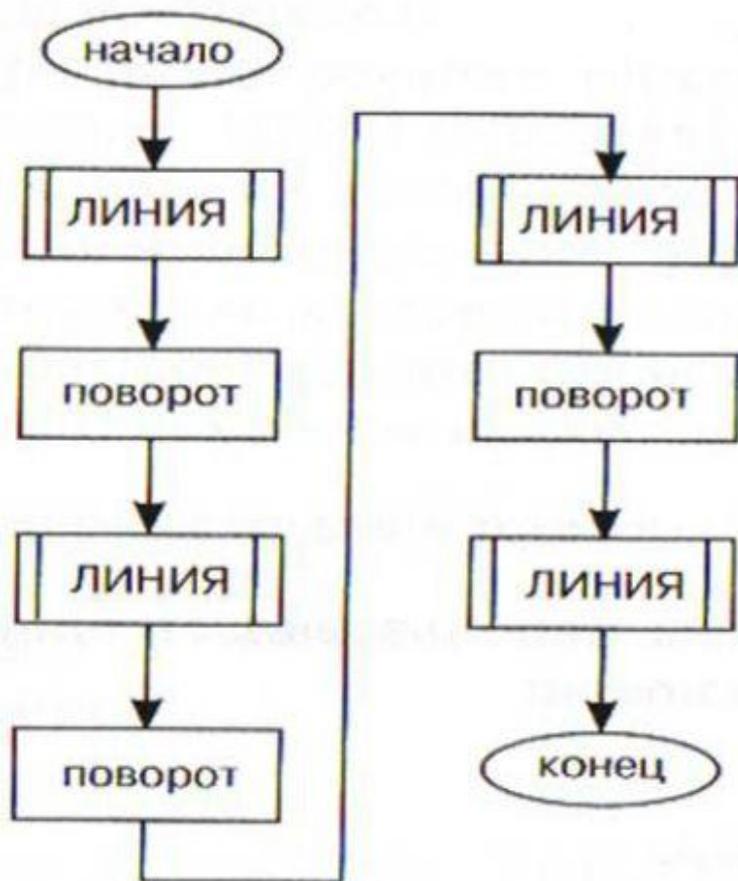
Цикл с постусловием

Тело цикла с постусловием всегда будет выполнено хотя бы один раз. Оно будет выполняться до тех пор, пока значение условного выражения ЛОЖНО. Как только условное выражение принимает значение ИСТИНА, цикл завершается.



Посмотрите на рисунок, где показана блок-схема алгоритма рисования рамки. Она состоит из двух частей: блок-схемы основного алгоритма и блок-схемы вспомогательного алгоритма ЛИНИЯ.

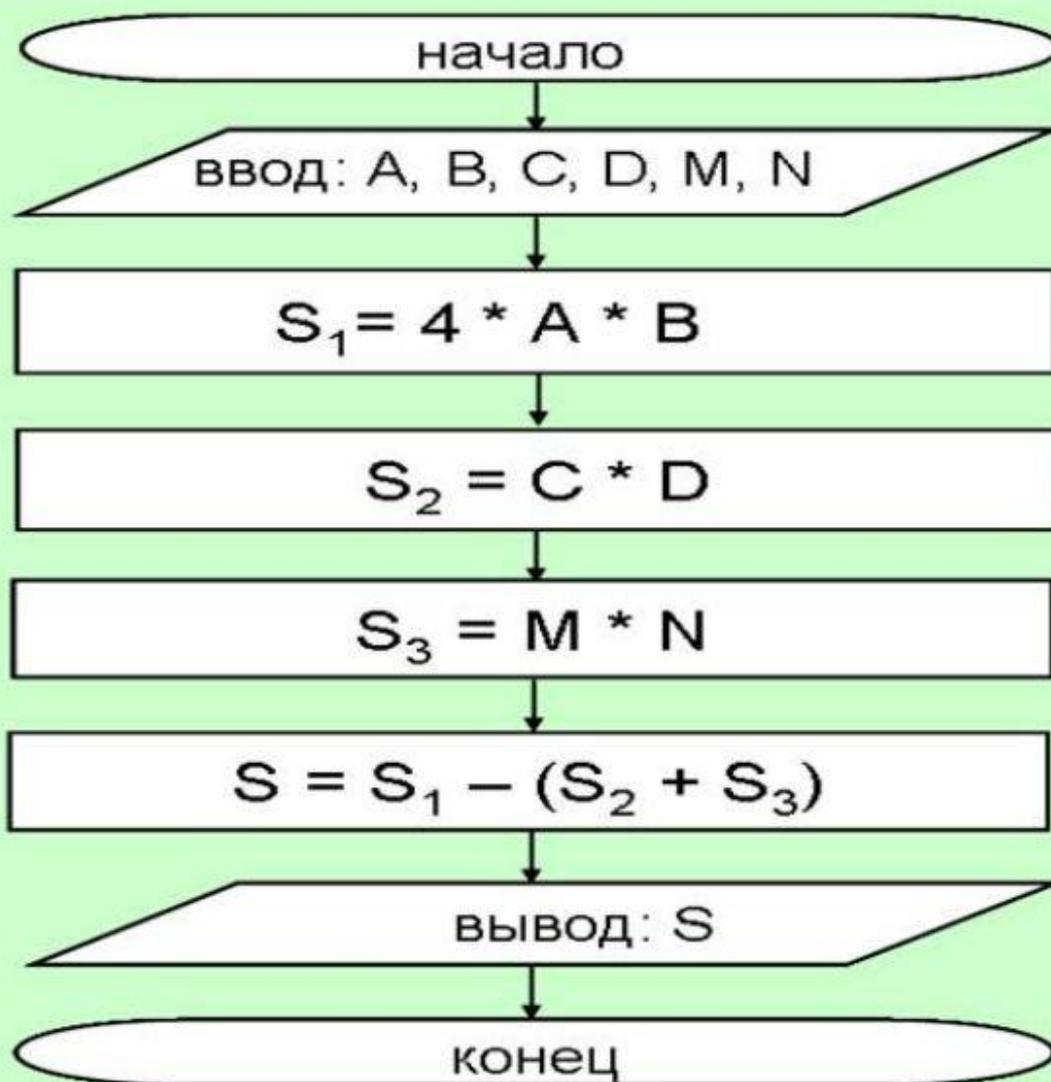
Основной алгоритм



Вспомогательный алгоритм



ПРИМЕР: В квадратной комнате шириной A и высотой B есть окно с размерами C на D и дверь M на N соответственно. Вычислите площадь стен для оклеивания их обоями. Блок-схема алгоритма решения поставленной задачи.

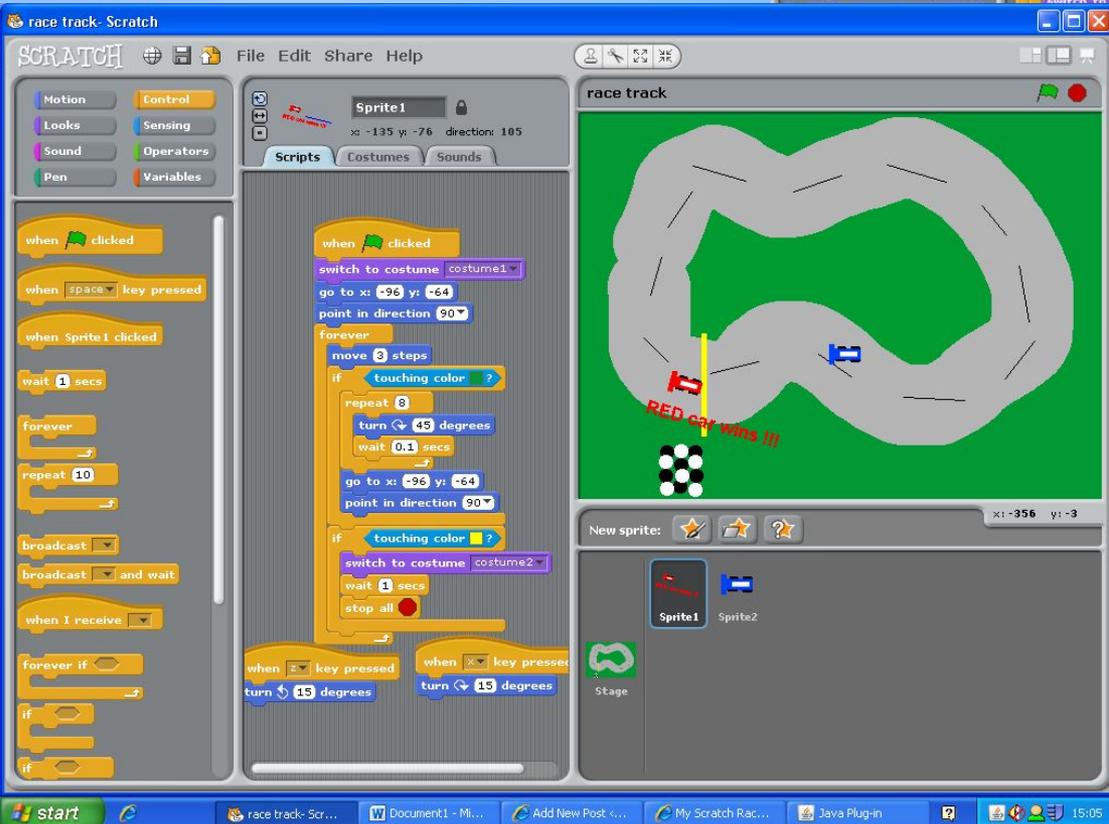




В LEGO WeDo блоки команд находятся в нижней части экрана.

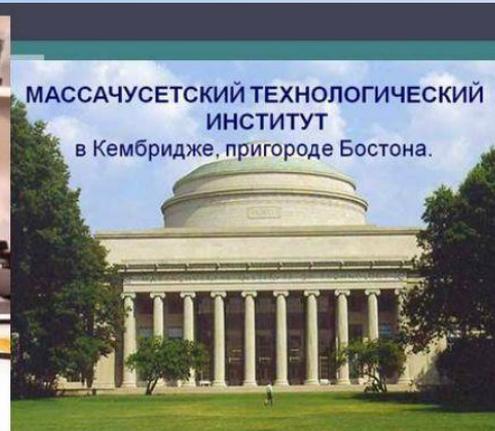
Применение у младших школьников «взрослых» языков программирования не является целесообразным, поскольку у детей еще слабые навыки абстрактного мышления, необходимые для полноценного программирования, т.е. лучше использовать языки и среды программирования, специально разработанные для обучения детей, с учетом психофизиологического и интеллектуального развития детей - *учебные языки программирования*, большинство из которых являются начальным или промежуточным звеном перед работой в средах программирования профессионального уровня.

SCRATCH



игры
мультфильмы
обучающие ролики

- **Scratch** («скрэтч») — это *визуальная объектно-ориентированная среда программирования для обучения детей, в которой блоки программ собираются из разноцветных кирпичиков-команд.*
- Название «Scratch» произошло от слова «scratching» — техники, используемой хип-хоп диджеями, которые крутят виниловые пластинки, чтобы смешать музыкальные темы.
- Среда Scratch разработана в 2007 на базе языка Squeak и поддерживается небольшой командой исследователей MIT Media Lab (возглавл. проф. Митчелл Резник) из Массачусетского технологического института (<http://scratch.mit.edu>).
- Scratch является бесплатным продуктом.



SCRATCH



Термины среды Скретч:

спрайт (персонаж), сцена (поле 480x360),
костюмы (вариант внешности),
скрипты (блоки команд для спрайта)

Файл ▾ Редактировать ▾ Помощь ▾

4_-_Tsvetok_de

Скрипты Костюмы Звуки

Движение События
Внешность Управление
Звук Сенсоры
Перо Операторы
Данные ScratchDuino

идти 10 шагов
повернуть на 15 градусов
повернуть на 15 градусов
повернуть в направлении 90
повернуться к
перейти в x: -48 y: -88
перейти в указатель мышки
плыть 1 секунд в точку x: -48 y: -88
изменить x на 10
установить x в 0
изменить y на 10
установить y в 0

когда щелкнут по
всегда
если Свет > 30, то
показать
сказать Привет!
иначе
сменить костюм на цветок 3
сказать Спокойной ночи!

когда клавиша пробел нажата
стоп все

Робот	Лаборатория
Аналог 0: 0	Аналог 0: 0
Аналог 1: 0	Аналог 1: 0
Аналог 2: 0	Аналог 2: 0
Аналог 3: 0	Аналог 3: 0
Аналог 4: 0	Аналог 4: 0
Кнопка: false	Свет: 0
	Звук: 0
	Рычажок: 0

Сцена 3 фоны
Новый фон:
Sprite1
цветок.2

The screenshot displays the Scratch 2.0 interface. At the top, there are menu options: 'Файл', 'Редактировать', and 'Помощь'. Below the menu is a toolbar with icons for saving, undo, redo, and help. The main workspace is divided into several panels:

- Top Left:** A preview window showing a museum gallery scene with a robot character. Below it, the coordinates are displayed as 'x: 240 y: -180'.
- Bottom Left:** A 'Робот' (Robot) panel with a table of sensor data:

Робот	Лаборатория
Аналог 0:	0
Аналог 1:	0
Аналог 2:	0
Аналог 3:	0
Аналог 4:	0
Кнопка:	false
Свет:	0
Звук:	0
Рычажок:	0
- Center:** A 'Палитра блоков' (Block Palette) with categories: Движение, Внешность, Звук, Перо, Данные, События, Управление, Сенсоры, Операторы, and ScratchDuino.
- Right:** A script editor showing a sequence of blocks:
 - 'когда щелкнут по флагу' (when green flag clicked)
 - 'установить x в 40' (set x to 40)
 - 'установить y в -163' (set y to -163)
 - 'всегда' (always) loop containing:
 - 'если Аналог Лаборатория 0 < 50, то' (if sensor 0 < 50, then) block with 'изменить x на -2' (change x by -2)
 - 'если Аналог Лаборатория 1 < 50, то' (if sensor 1 < 50, then) block with 'изменить y на 2' (change y by 2)
 - 'если Аналог Лаборатория 2 < 50, то' (if sensor 2 < 50, then) block with 'изменить x на 2' (change x by 2)
 - 'если Аналог Лаборатория 4 < 50, то' (if sensor 4 < 50, then) block with 'изменить y на -2' (change y by -2)
 - 'если на краю, оттолкнуться' (if touching edge, bounce)
 - 'стиль вращения влево-вправо' (rotation style: left-right)
- 'когда клавиша пробел нажата' (when space key pressed) block with 'стоп все' (stop all)
- 'когда щелкнут по флагу' (when green flag clicked) block with 'всегда' (always) loop containing:
 - 'следующий костюм' (next costume)
 - 'ждать 1 секунд' (wait 1 seconds)

При создании скрипта (программы) в Scratch используется палитра блоков, которая занимает центральную часть экрана. В ее верхней части располагается 8 разноцветных кнопок, которые выбирают нужную группу команд. Команды выбранной группы отображаются в окне сверху вниз.

Как управлять колёсной робоплатформой



в программе RobboScratch?



Скрипты Костюмы Звуки

Движение Внешность Звук Перо Данные РобоТ

События Управление Сенсоры Операторы Новые блоки Лаборатория

моторы вкл на 1 секунд

моторы вкл

моторы выкл

уст направление РобоТ вперед

моторы вкл на 10 шагов

обнулить счётчик пути

повернуть на 15 градусов

повернуть на 15 градусов

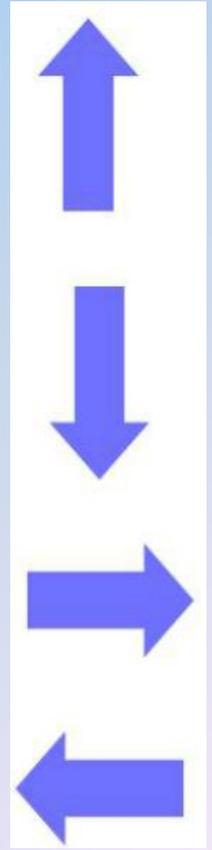
Задаем направление движения.

уст направление РобоТ вперед

уст направление РобоТ назад

уст направление РобоТ направо

уст направление РобоТ налево



Анимация в среде программирования Scratch



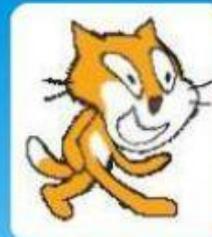
- * Трудно представить компьютерный проект без элементов анимации. Анимация (от латинского «*anima*» - душа) - это оживление изображений быстрой сменой неподвижных картинок.
- * Основа алгоритма анимации - это повтор (цикл) нескольких действий. Можно сделать разные варианты алгоритма анимации в зависимости от задачи автора.

Алгоритм анимации



- * У спрайта есть костюмы. Быстрой сменой костюмов можно анимировать движение спрайта. Чем больше костюмов, изображающих моменты движения, тем реалистичнее будет анимация.

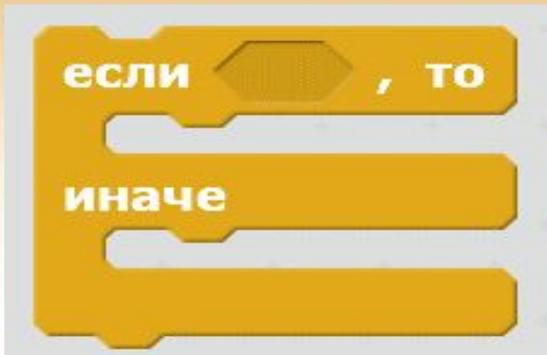




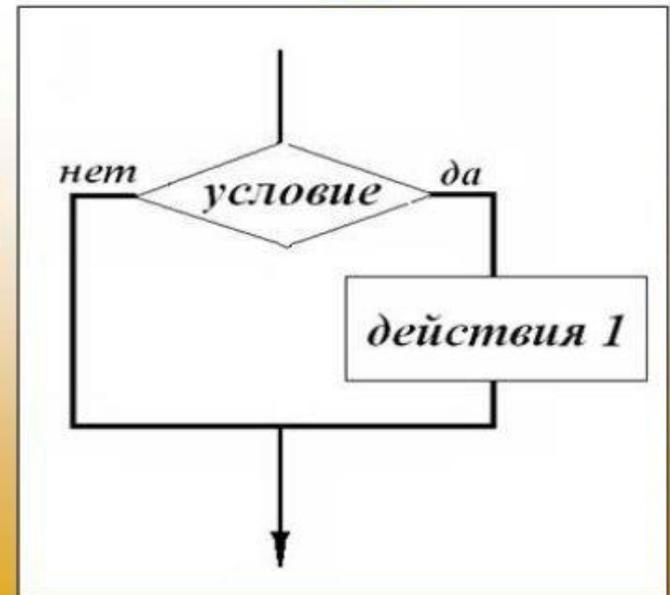
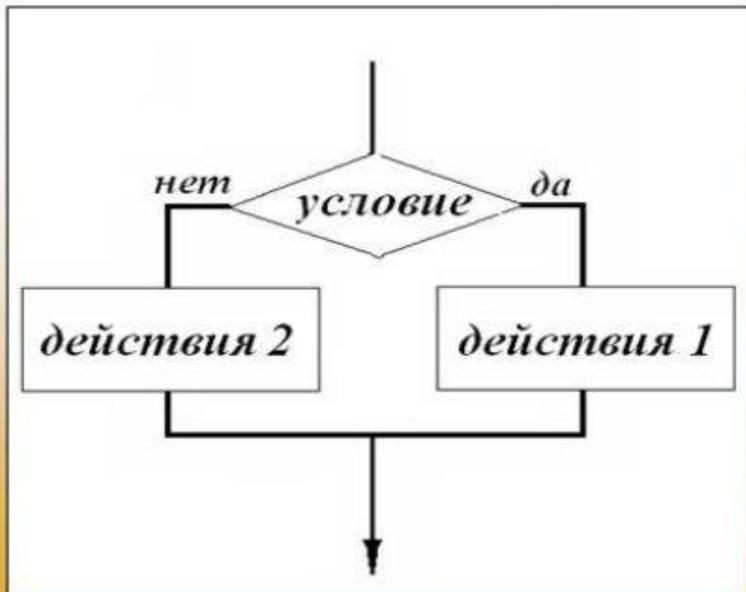
- * Для усиления зрелищности можно добавить графический эффект из категории внешность, например, "завихрение" как подобие вдоха и выдоха спрайта. В жизни размер удаляющегося от нас объекта кажется все меньше и меньше. В Скретче с каждым шагом мы тоже можем уменьшать размер спрайта вплоть до 0%.

```
когда щелкнут по [флаг]
всегда
  поставить фон в [выдать случайное от 1 до 3]
  если фон = 1
    изменить завихрение эффект на [выдать случайное от -50 до 50]
  или
    если фон = 2
      изменить рыбий глаз эффект на [выдать случайное от -50 до 50]
    или
      установить эффект цвет в значение [выдать случайное от 0 до 50]
  ждать 5 секунд
```

Алгоритмические структуры и команды языка Scratch



Выбор:



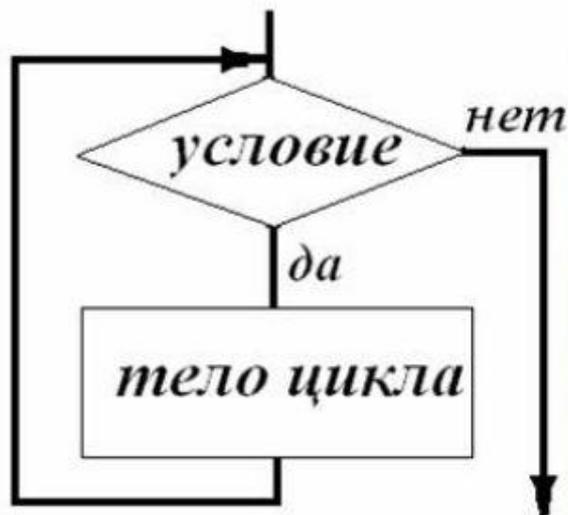
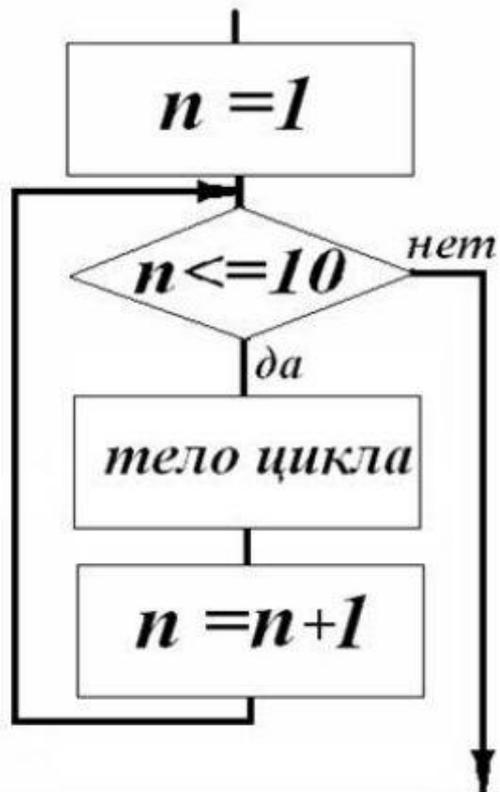
повторить 10



Повторы:



повторять пока не



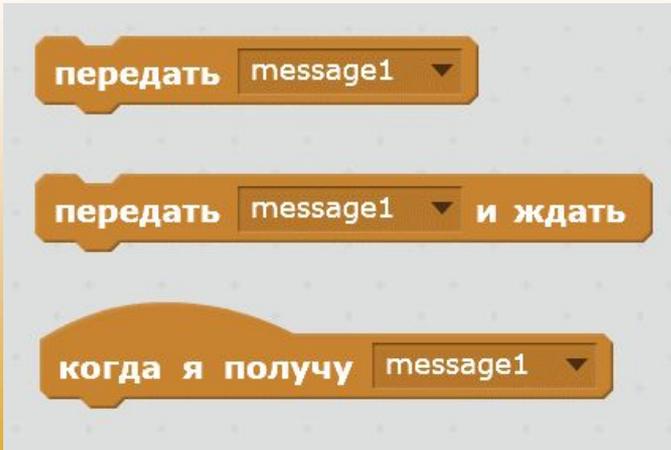
всегда



Для автоматической смены действий удобно использовать такого «дирижера», как

Механизм передачи сообщений

Один объект посылает сообщение всем остальным объектам, которые реагируют на это сообщение определенным образом или оставляют его без внимания.

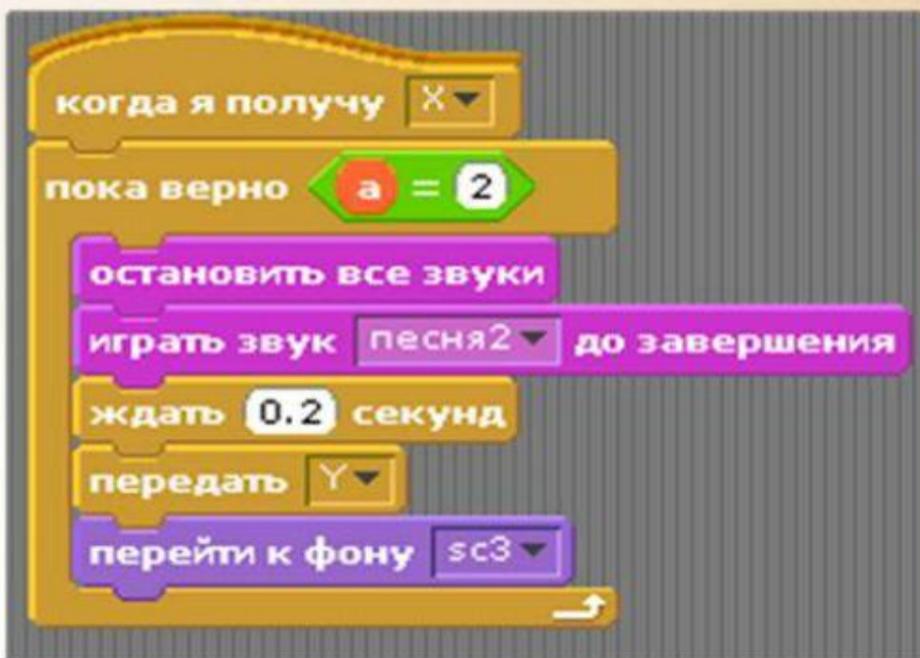


передать message1

передать message1 и ждать

когда я получу message1

Примеры передачи сообщений



Примеры заданий для детей 6–10 лет

1 этап при освоении любой команды (скрипта) программирования – бескомпьютерный.

ЗАДАНИЯ:

- составь/зарисуй алгоритм игры, складывания оригами, изготовления шоколада, выпечки печенья, пришивания пуговицы и т.п.;
- зарисуй алгоритм перемещения по клеткам, используя стрелки;
- соедини элементы паззла вместе, чтобы получилась картинка



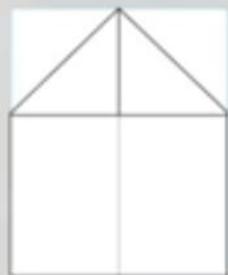
CUT CENTER OUT OF PAPER



CREASE PAPER DOWN THE CENTER



CRUMBLE PAPER



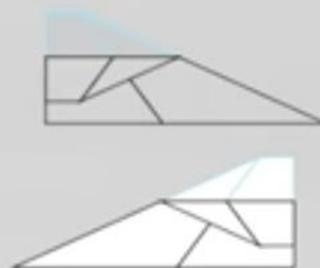
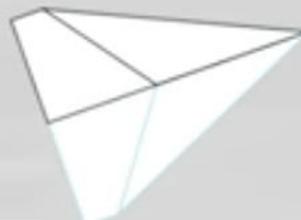
FOLD TOP CORNERS TO CENTER

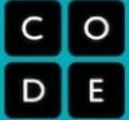


RIP CORNER OFF PAPER



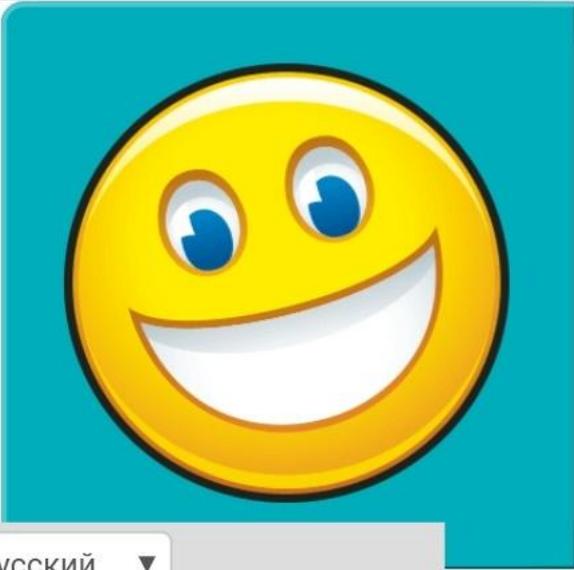
FOLD CORNER SIDES TO CENTER



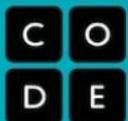


Перетащи блок к цели

Место сбора блоков:



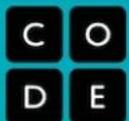
Русский ▾



Собери картинку из блоков

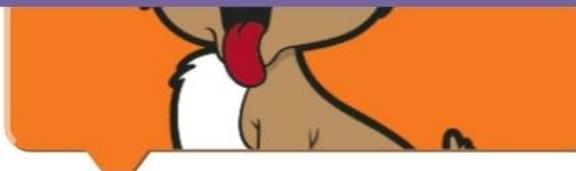
Место сбора блоков:





Собери картинку из блоков

Место сбора блоков:

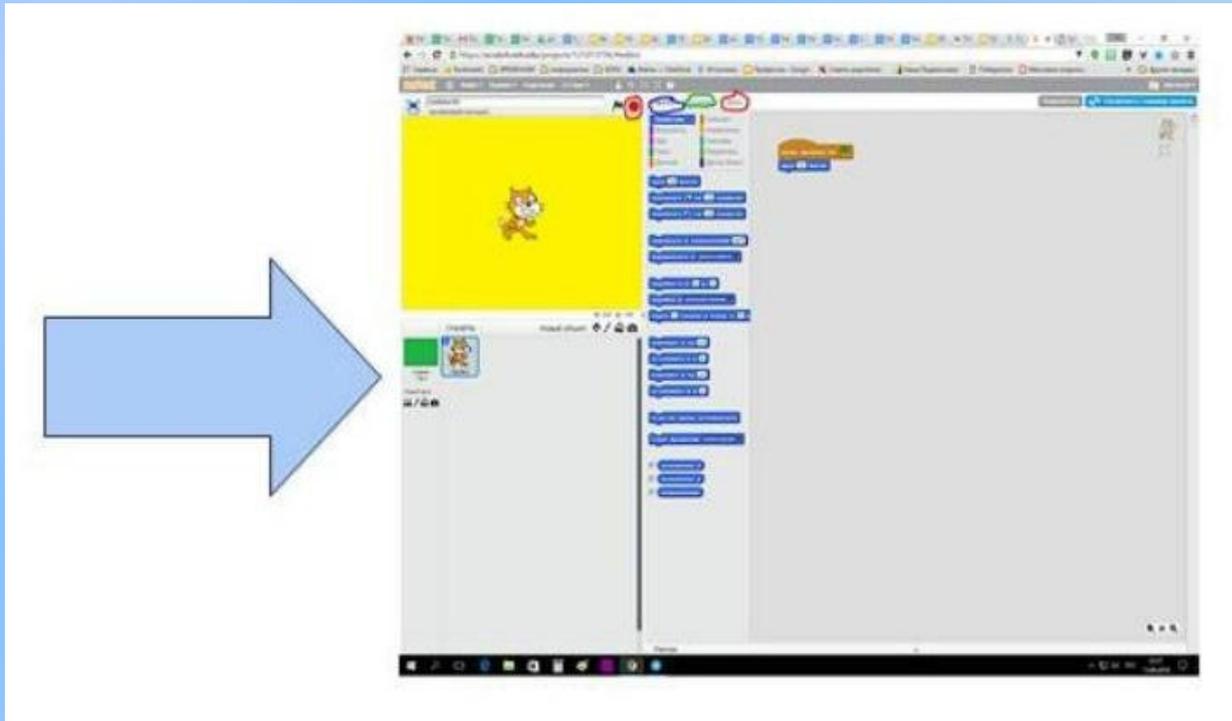




Пример из мобильного приложения Coding for kids

ЗАДАНИЕ в рабочей тетради:

- Раскрасьте желтым карандашом место, где происходит действие программы;
- Раскрасьте зеленым карандашом сцену(фон);
- Обведите синим карандашом вкладку скрипты;
- Обведите зеленым карандашом вкладку костюмы



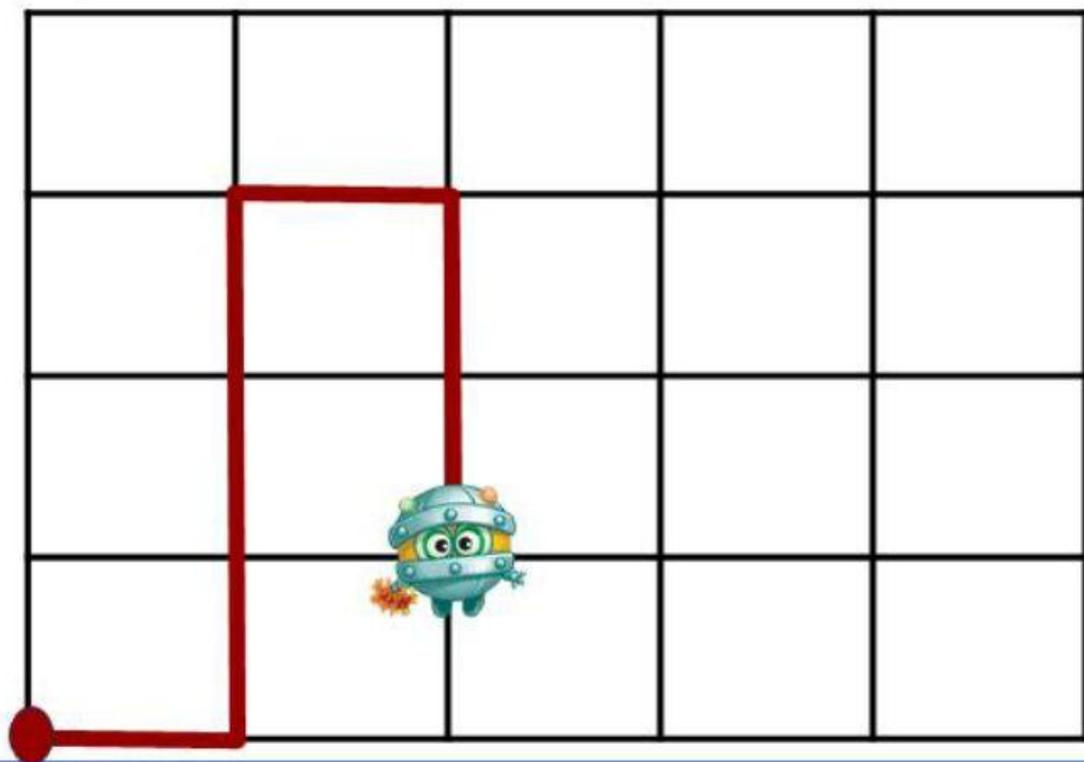
*Пример из
RobboScratch*

ЗАДАНИЕ в рабочей тетради:

- Обведи красным карандашом сигнал «остановить выполнение файла». Какой он формы, какого цвета?
- Какой еще значок нарисован рядом?
Что он означает?

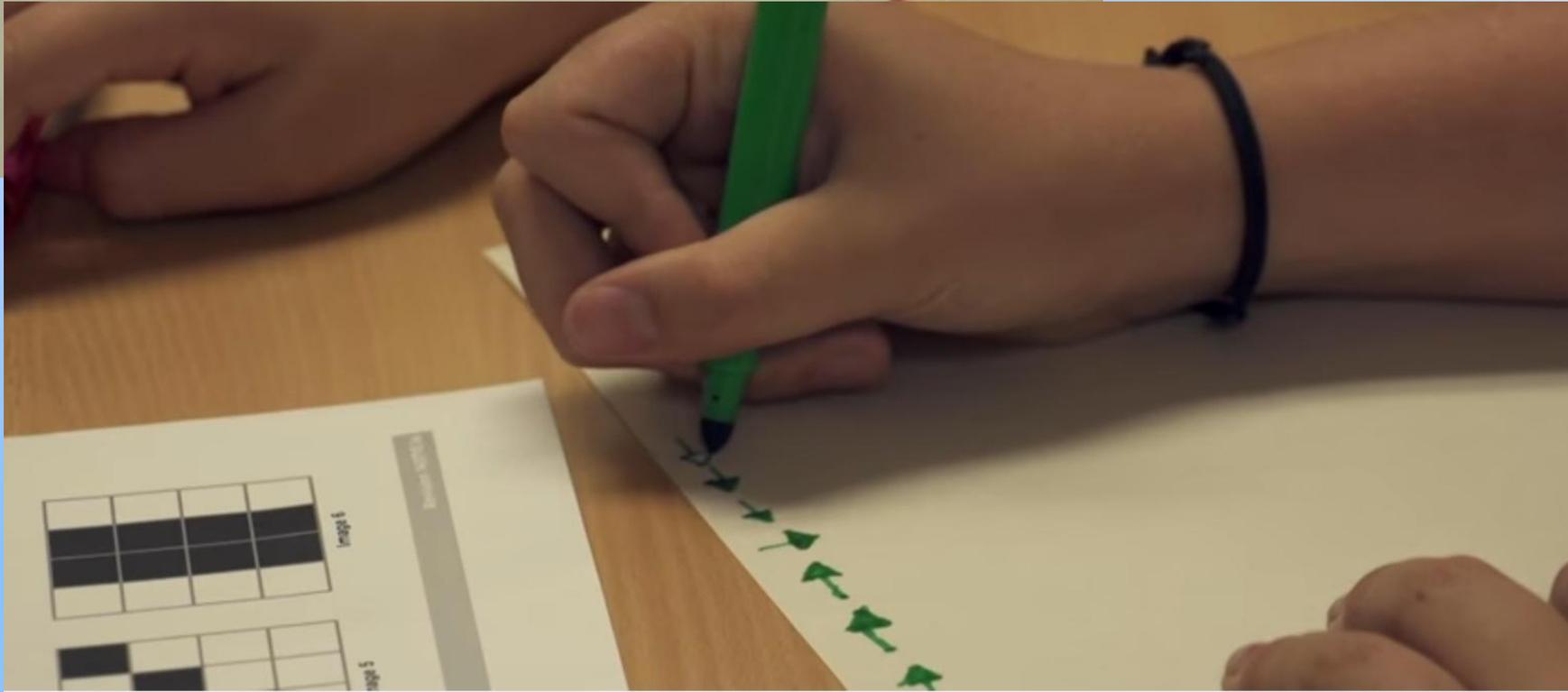
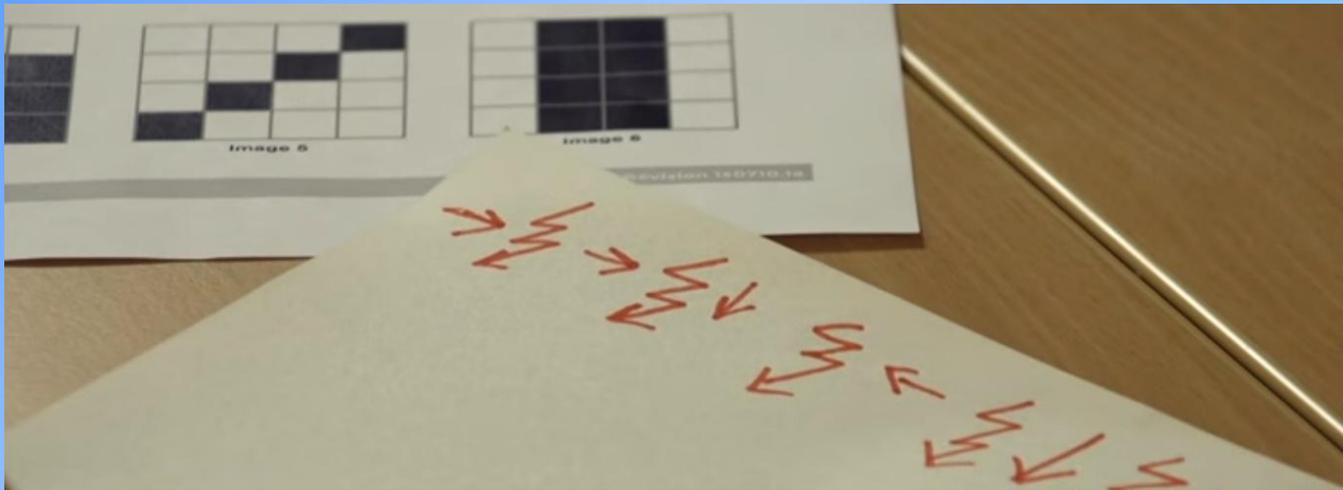


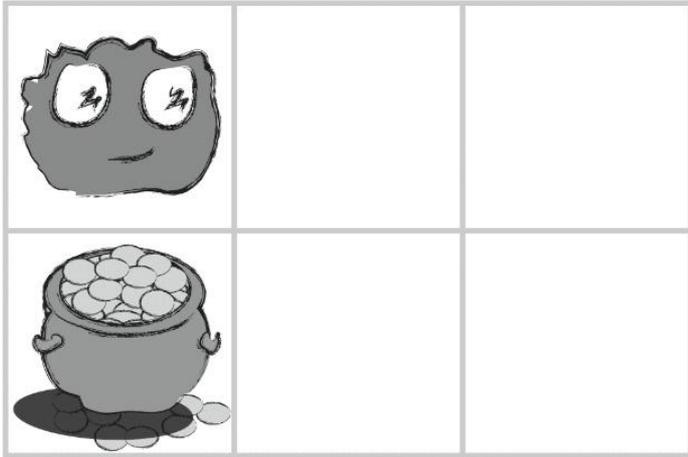
Пример из RobboScratch



Робот должен проехать по траектории.

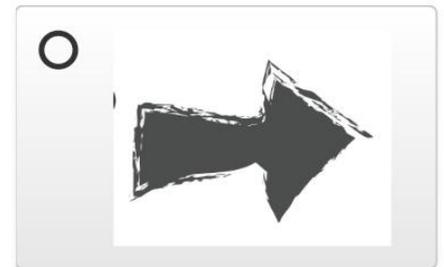
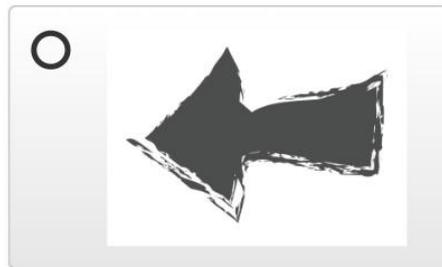
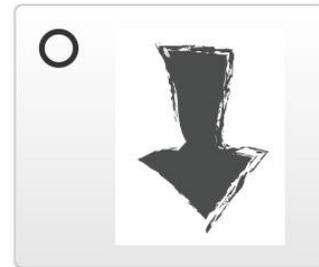
Составьте программу движения.





Какая стрелка направит Флурба к сокровищу?

Отправить



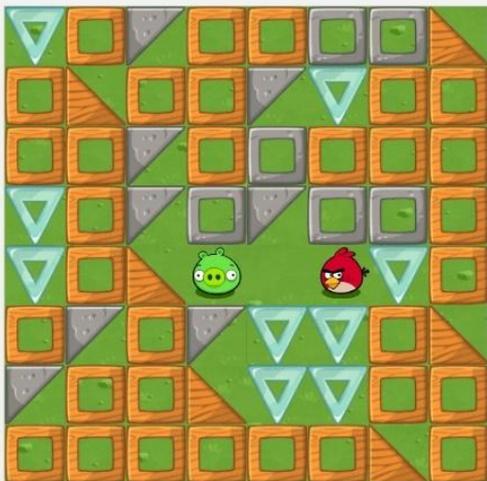
https://studio.code.org/s/course1/stage/4/puzzle/1



Этап 4: Лабиринт: Последовательность

1

ЕЩЁ



Поможешь мне двигаться на запад, чтобы добраться до свињи?

Блоков

Место сбора блоков: 1 / 3 блоки

Начать заново



при запуске ▶



▶ Выполнить

Русский ▼

Лабиринт

С помощью какого ответа можно пройти этот уровень?

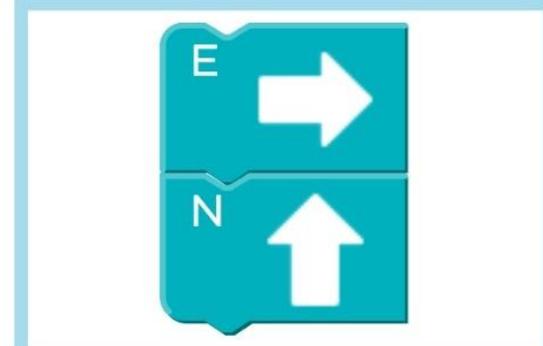
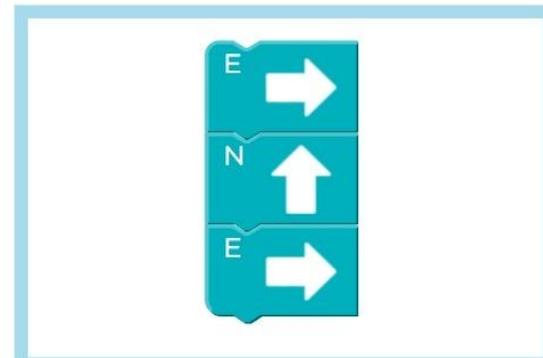
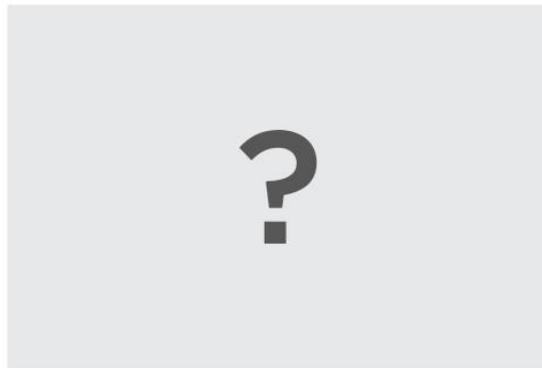
Отправить

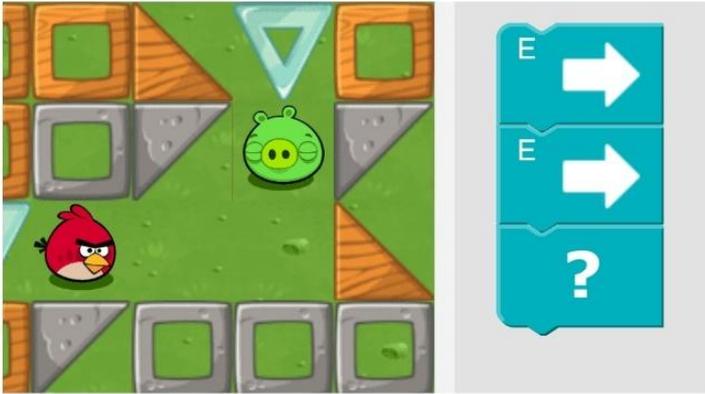


Соответствие

Сопоставь головоломки и блоки

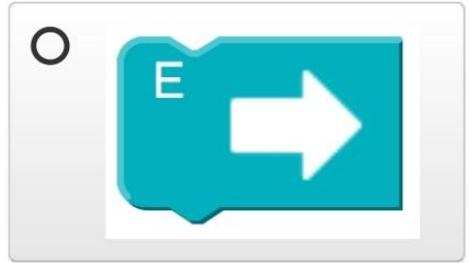
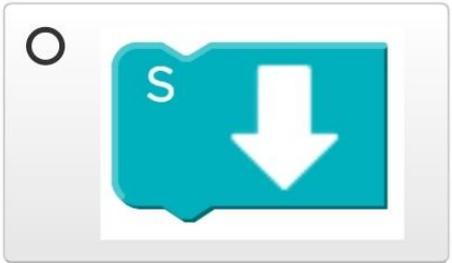
Отправить





Какой блок должен быть последним?

Отправить





▶ Выполнить

шаг

Русский ▼

[Политика конфиденциальности](#) | [At](#)



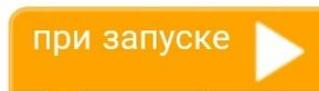
Отладь этот уровень, переставляя и изменяя блоки, чтобы довести меня до свиньи!

Блоков

Место сбора блоков: 9 / 6 блоки

↻ Начать заново

</> По



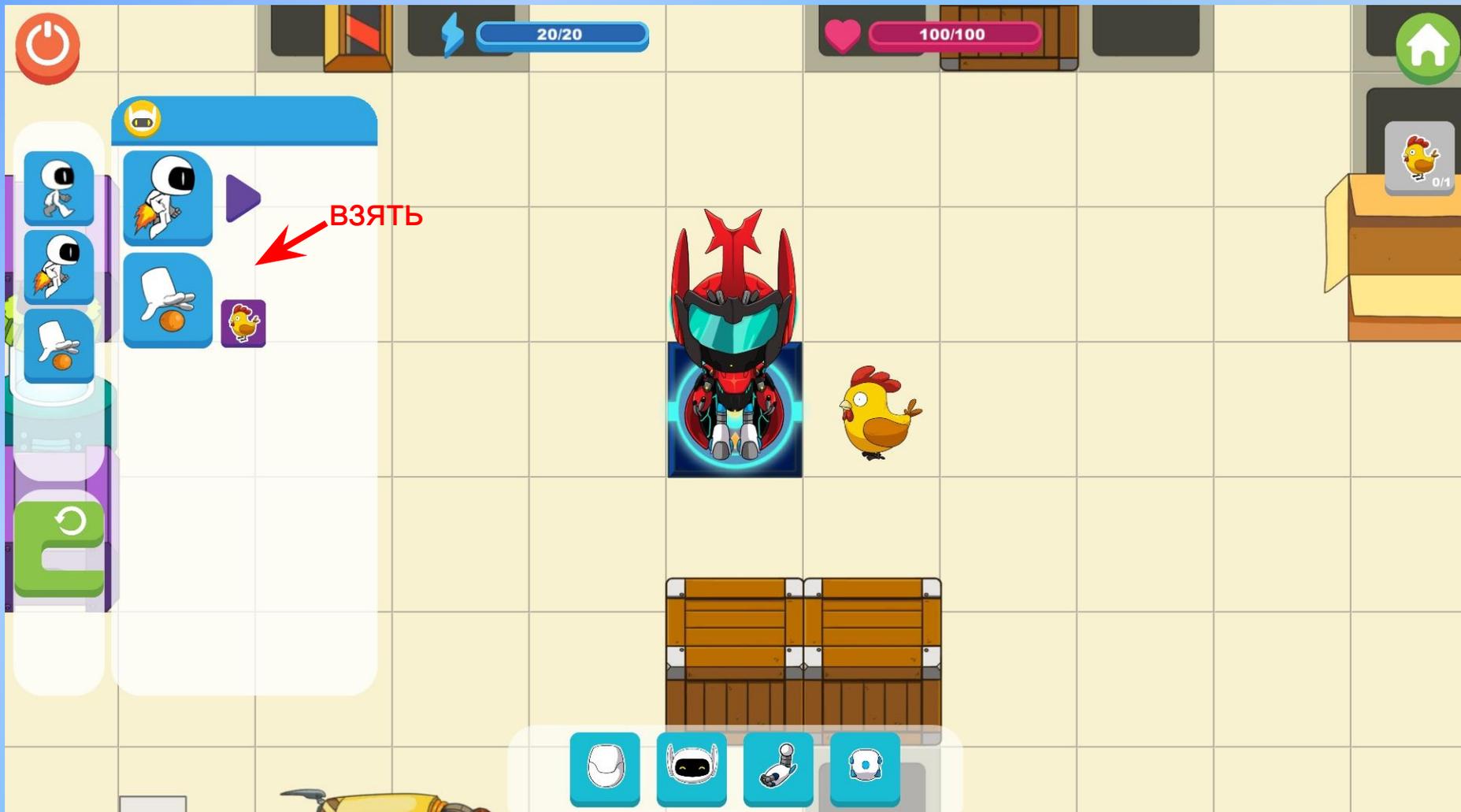
Усложнение – к командам движения добавляются другие команды, команды используются в словесной форме



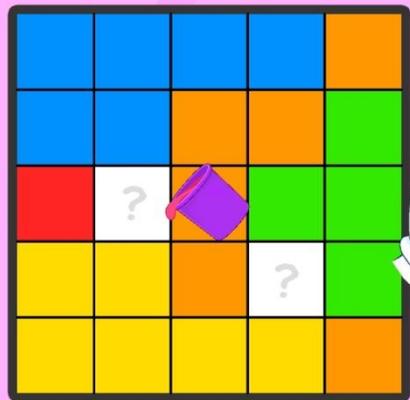
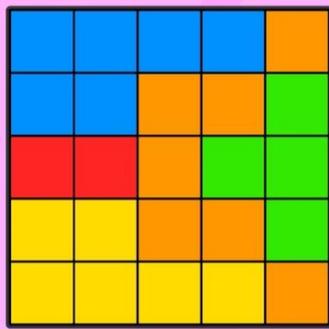
Пример из мобильного приложения Coding for kids



Пример из
мобильного
приложения
Coding for kids



Пример из мобильного приложения Robotizen



Control panel with various icons:

- Right arrow
- Down arrow
- Blue block with drop icon
- Left arrow
- Right arrow
- Left arrow
- Blue block with drop icon
- Up arrow



Перемести меня к цветку, собери нектар, а затем перемести меня к соте и приготовь мед

Блоков

Место сбора блоков: 5 / 6 б [Начать заново](#)

ВЗЯТЬ

сохранить

при запуске

В

ВЗЯТЬ

В

Выполнить

шаг

Русский ▾

ЗАДАНИЕ «Художник»: создай программу и соедини точки линиями

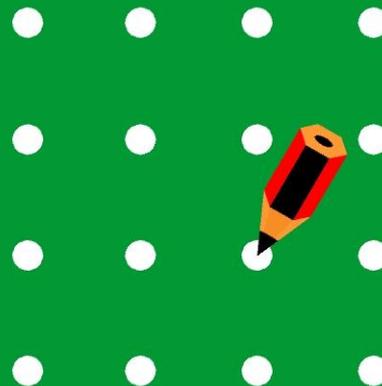
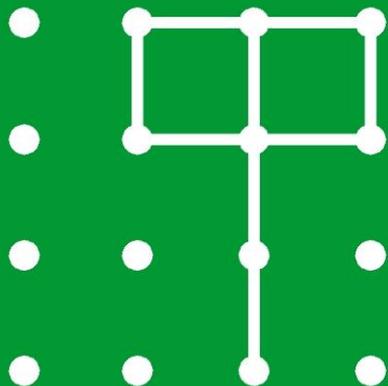


Пример из мобильного приложения Coding for kids



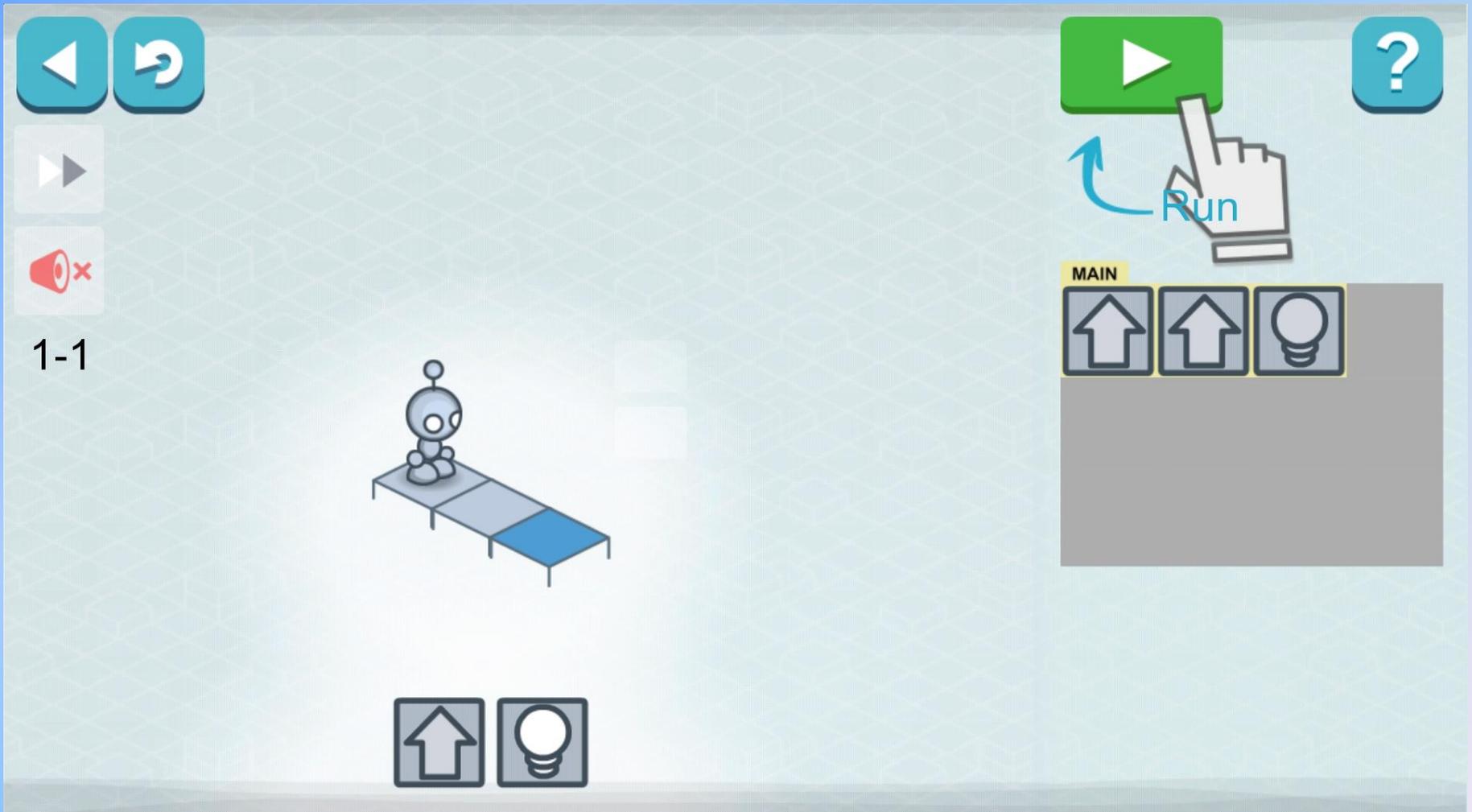
Level 8

64%  15:38



Go

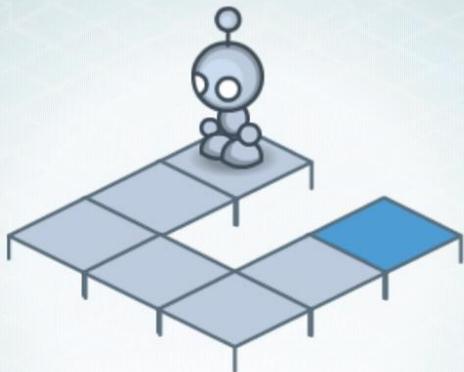




Пример из мобильного приложения Lightbot Hour



1-2

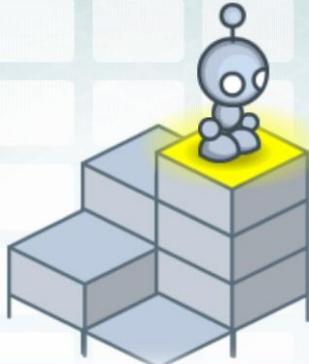


MAIN





1-3



MAIN





Можешь помочь мне поймать непослушную свинью?
Составь вместе несколько блоков «двигаться вперед»
и нажми «Выполнить».

Блоков

Место сбора блоков:



Начать заново

</> По

двигаться вперед

при запуске

повернуть налево ↻ ▾

повернуть направо ↻ ▾

▶ Выполнить

шаг

Русский ▾

Пример с сайта code.org



```
1 turn left
2 step 12
```

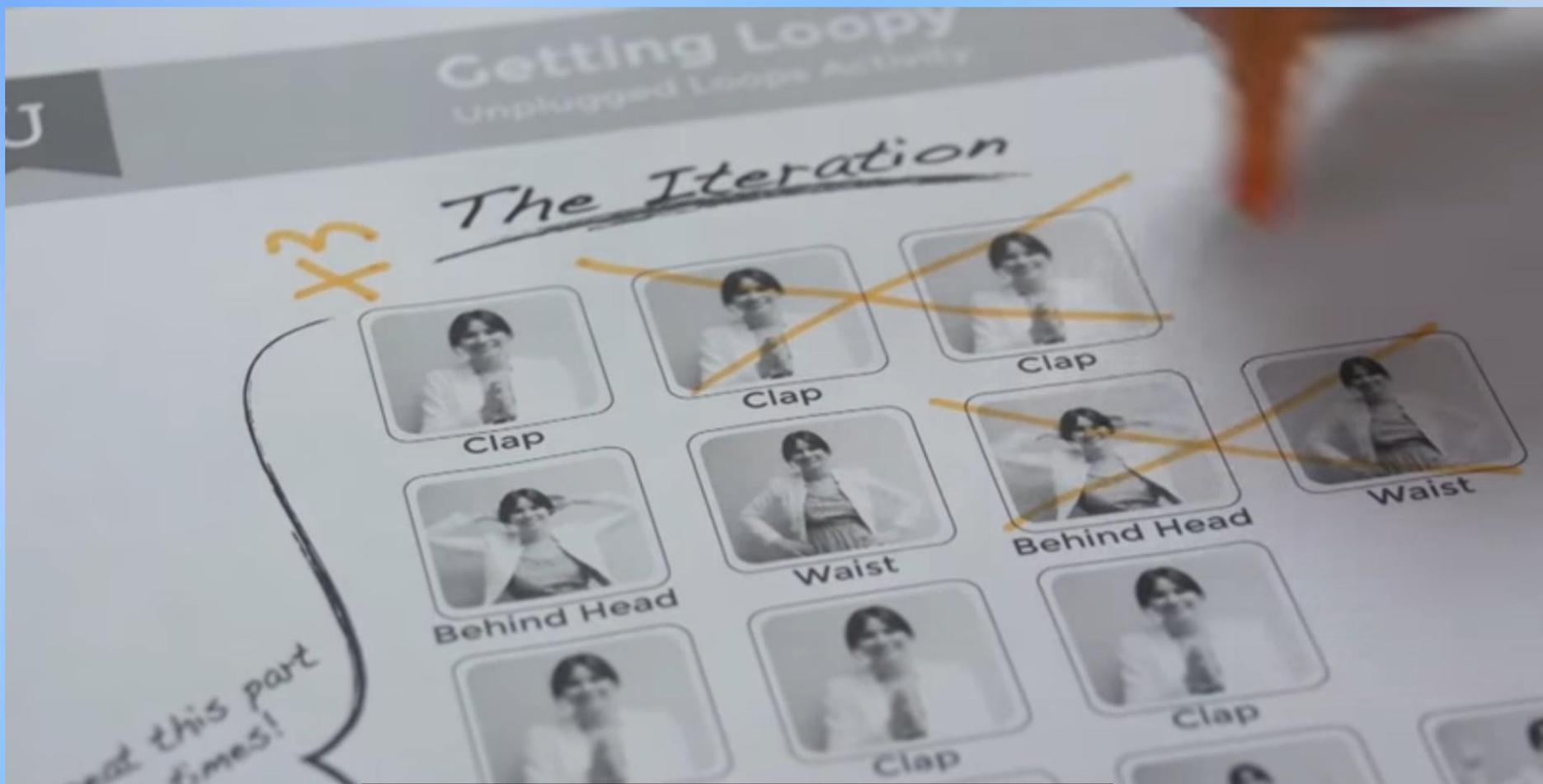
Используй **turn left** (повернуть налево) или **turn right** (повернуть направо), чтобы повернуть обезьяну к банану.

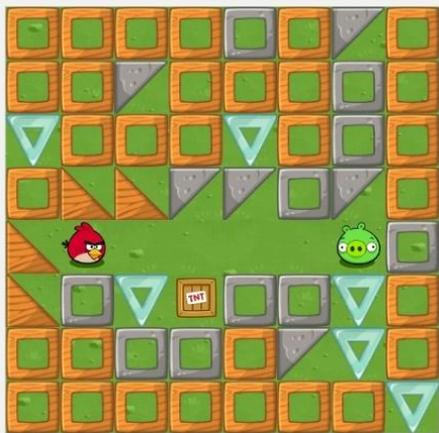


```
1 turn left
2 step 12|
```



3 этап: освоение понятия «цикл», формирование умений строить циклические алгоритмы





▶ Выполнить

шаг



Можешь довести меня до свиньи, используя 5 блоков?

Блоков

Место сбора блоков: Начать заново

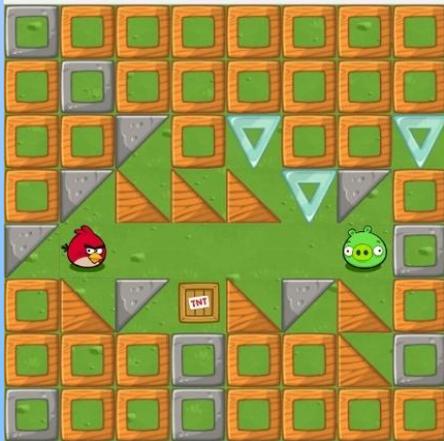
двигаться вперед

при запуске

повернуть налево

повернуть направо

Русский ▼



▶ Выполнить

шаг



Можешь заменить эти пять блоков только двумя блоками? (Подсказка: Чтобы пройти этот уровень, тебе нужно создать цикл, используя блок «повторить».)

Блоков

Место сбора блоков: Начать заново

1 двигаться вперед

при запуске

повернуть налево

повернуть направо

Русский ▼

Робоплатформа должна ехать 3 секунды.
Какая программа подойдёт?

включить мотор на 1 сек

включить мотор на 1 сек

включить мотор на 1 сек

включить мотор на 3 сек

Усложнение – использование цикла в цикле



Доведи меня до подсолнуха! (Подсказка: Чтобы пройти этот уровень, тебе нужно использовать один цикл «повторить» внутри другого.)

Блоков

Место сбора блоков:

 Начать заново



двигаться вперед

при запуске

повернуть **налево** ↻ ▼

повернуть **направо** ↻ ▼

 Выполнить

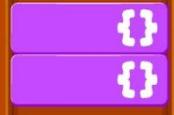
шаг

Русский ▼

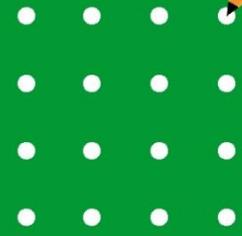
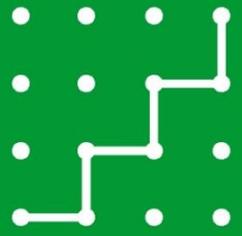
[Политика конфиденциальности](#) | [Автоматическое обновление](#)

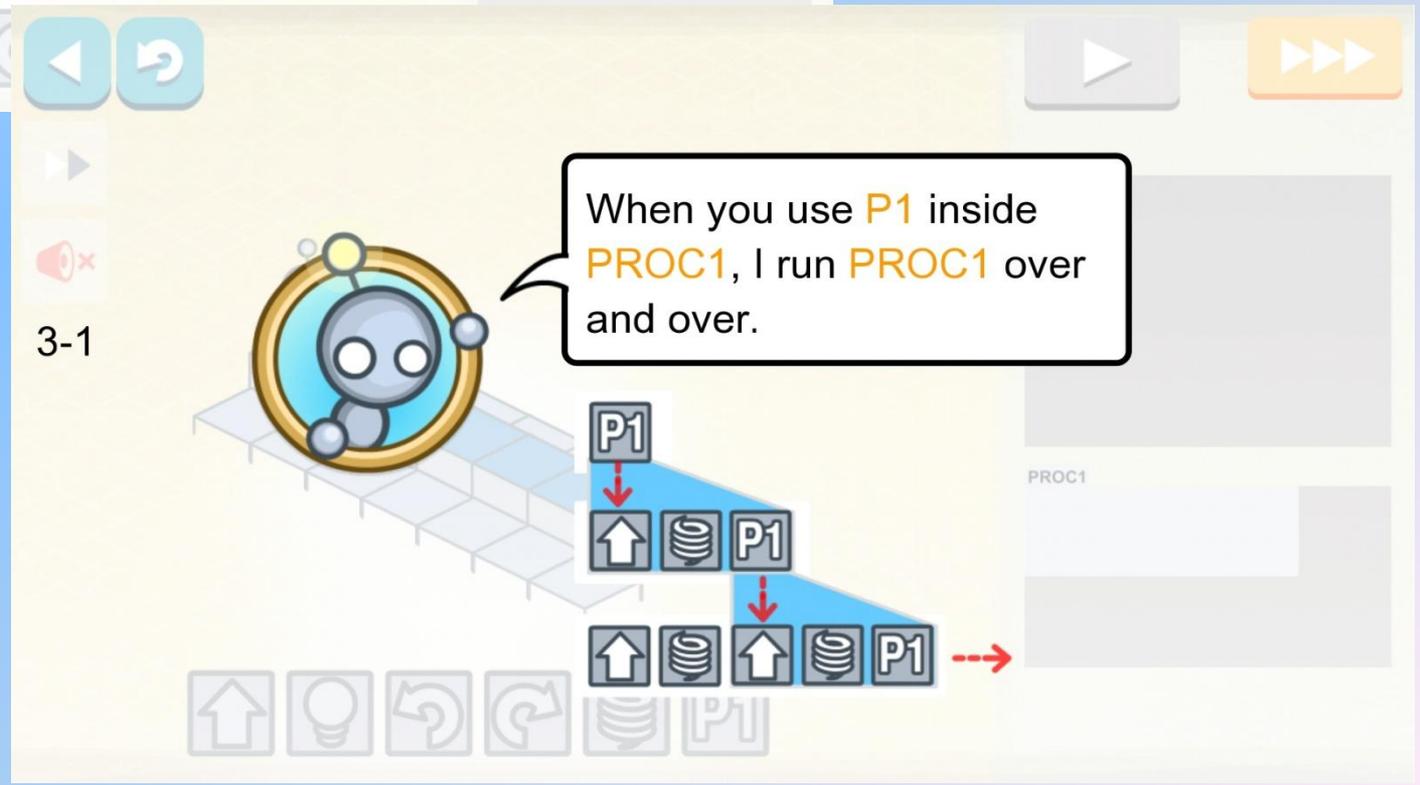
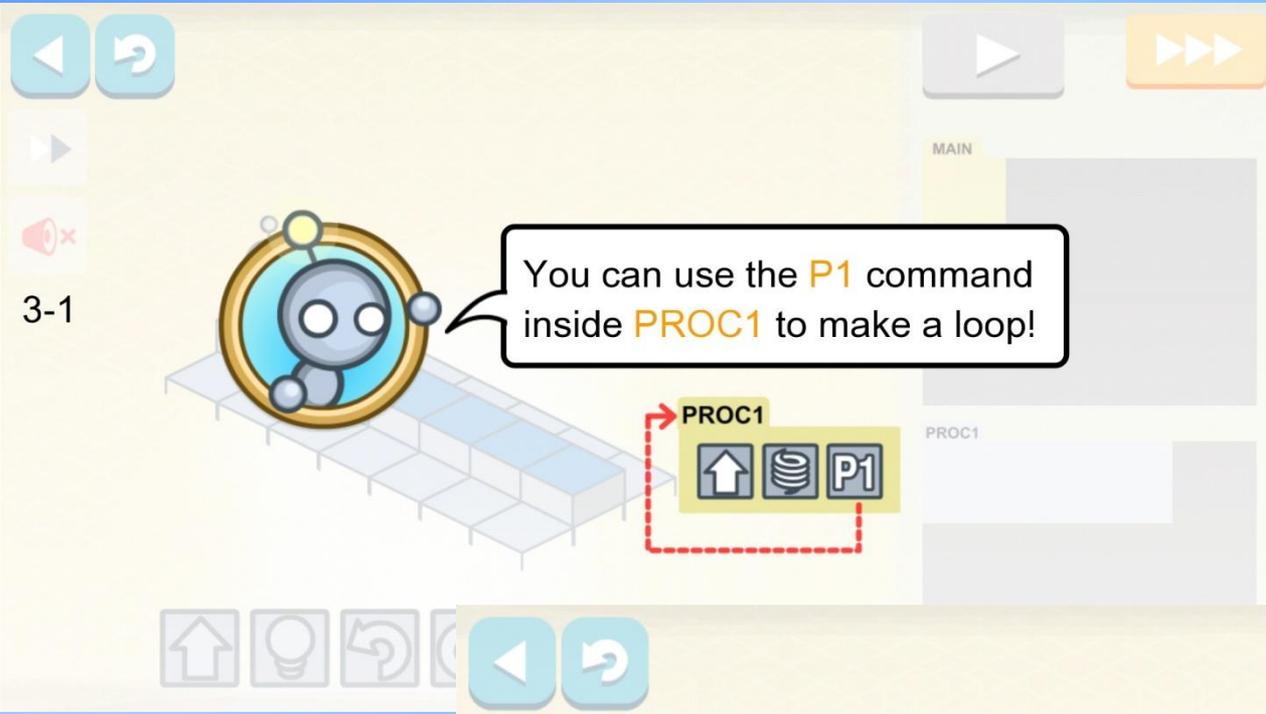


Level 1



Пример из мобильного приложения Coding for kids

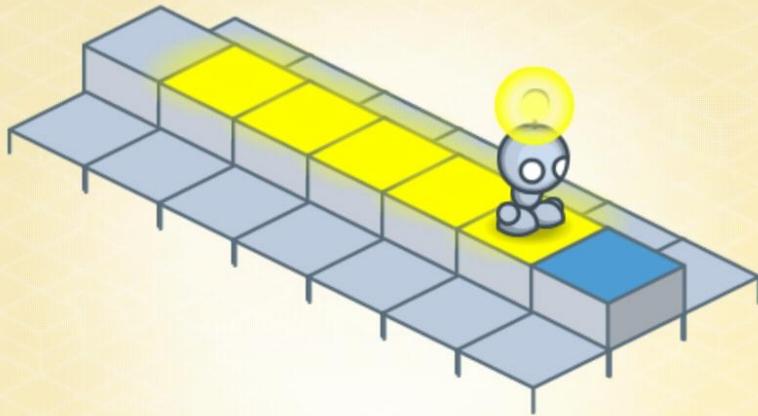




Пример из
мобильного
приложения
Robotizen



3-1



MAIN

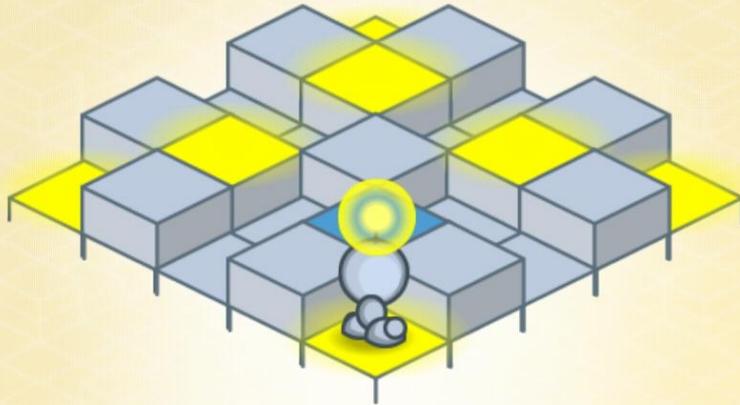


PROC1

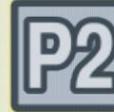




3-6



MAIN



PROC1

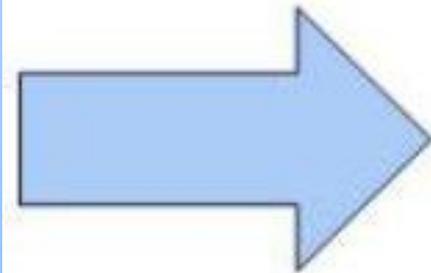


PROC2



Практическое задание в программе:

Перед вами спрайт - котенок. Давайте заставим его двигаться. Нажмите несколько раз на флажок. Что происходит?

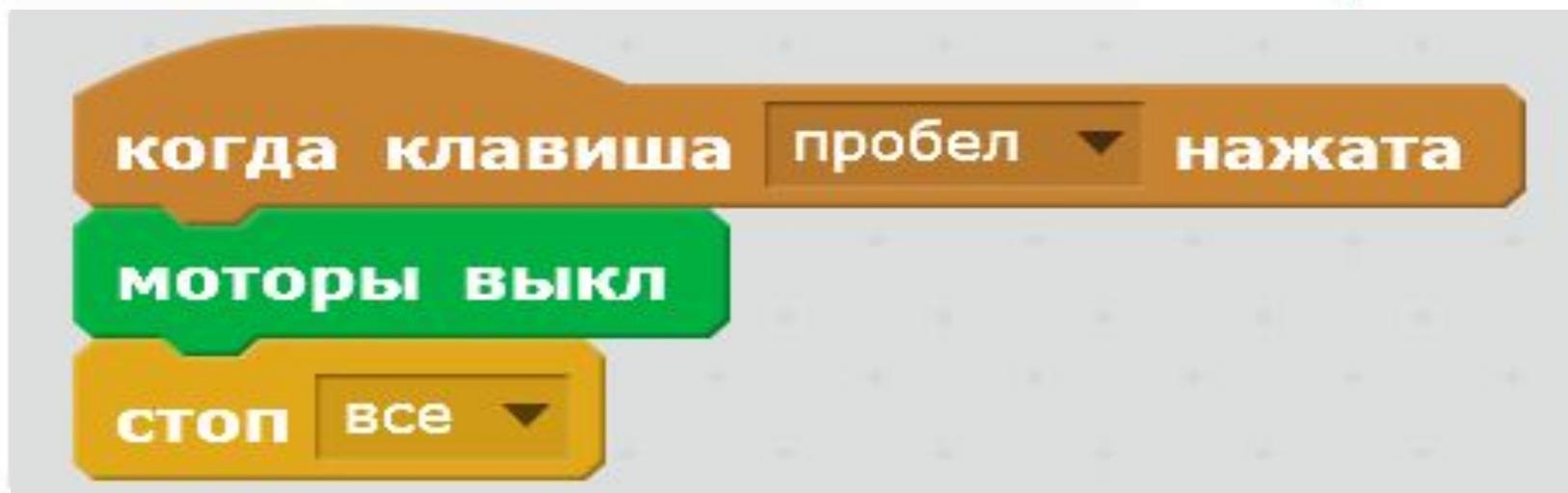


Пример из RobboScratch

4 этап – программирование движения робота



Важно: создать блок – тормоз!

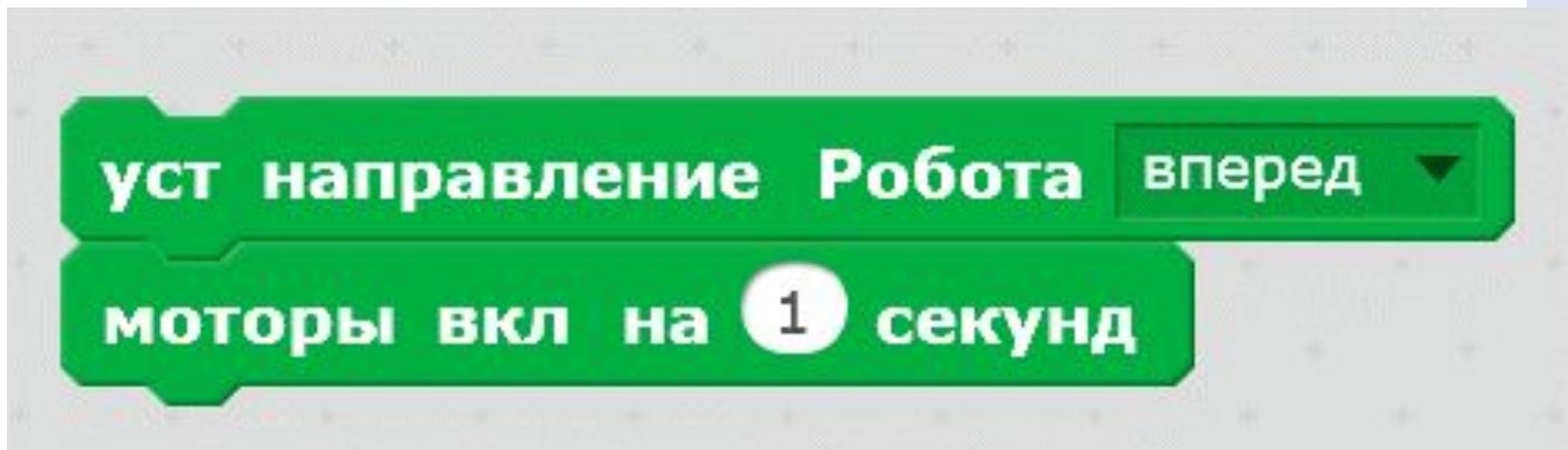


Пример из RobboScratch

Чего не хватает в программе? Создайте подобную программу, но поменяйте направление движения на «туда», понаблюдайте за движением робота.

Управление движением Робоплатформы:

1. Указать направление движения моторов
2. Включить мотор на некоторое время

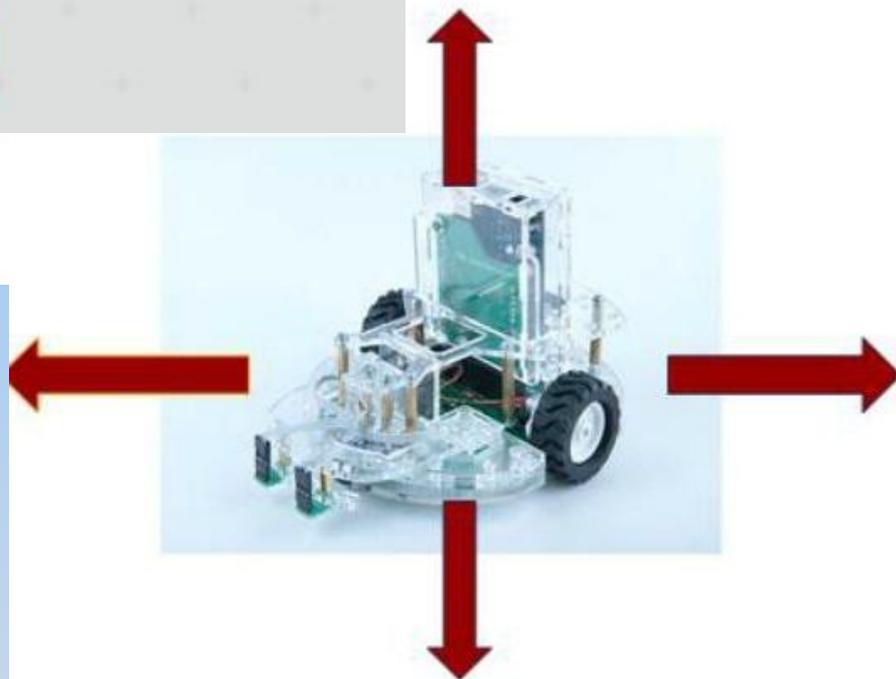


Как будет двигаться Робоплатформа при выполнении такой программы?

когда клавиша нажата

уст направление Робота

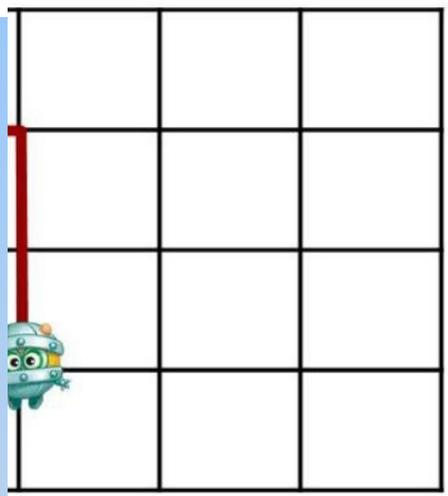
моторы вкл на секунд



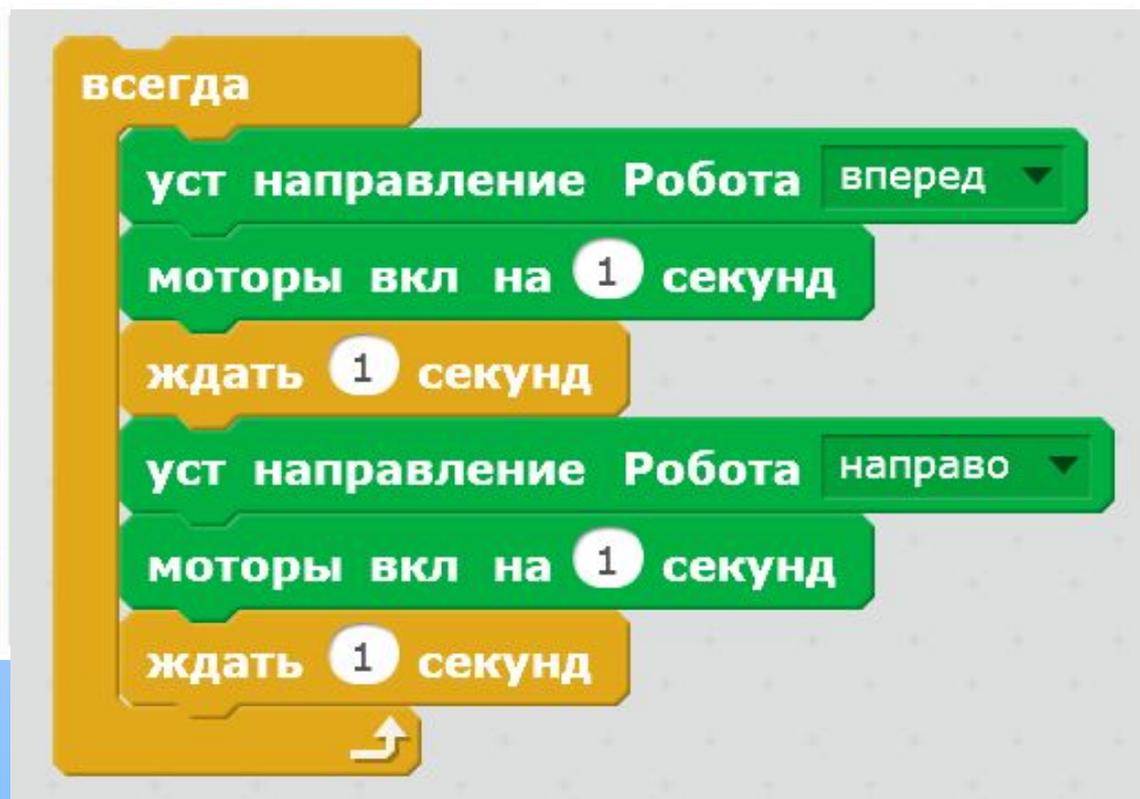
Сколько раз будет повторяться алгоритм? Где окажется робот? Поясни свой ответ.

```
уст направление Робота вперед
моторы вкл на 1 секунд
уст направление Робота направо
моторы вкл на 1 секунд
```

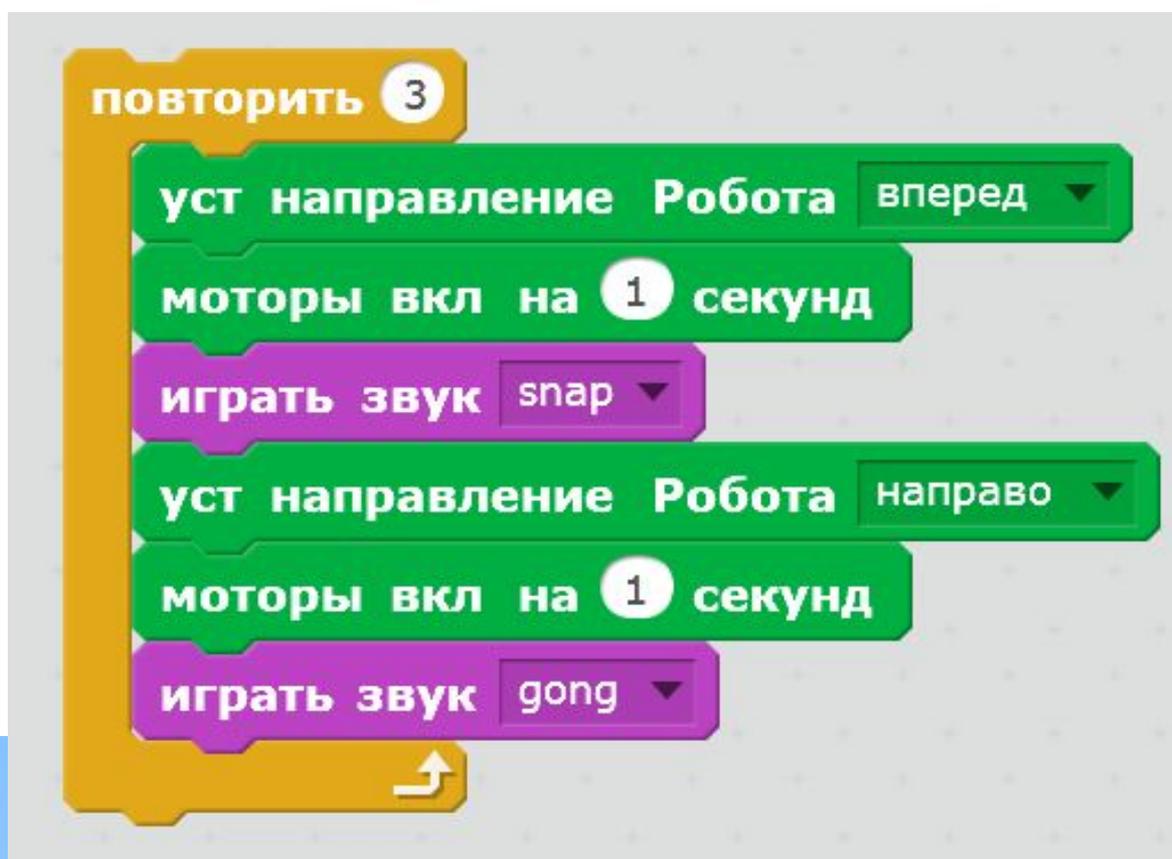
```
всегда
  уст направление Робота вперед
  моторы вкл на 1 секунд
  уст направление Робота направо
  моторы вкл на 1 секунд
```



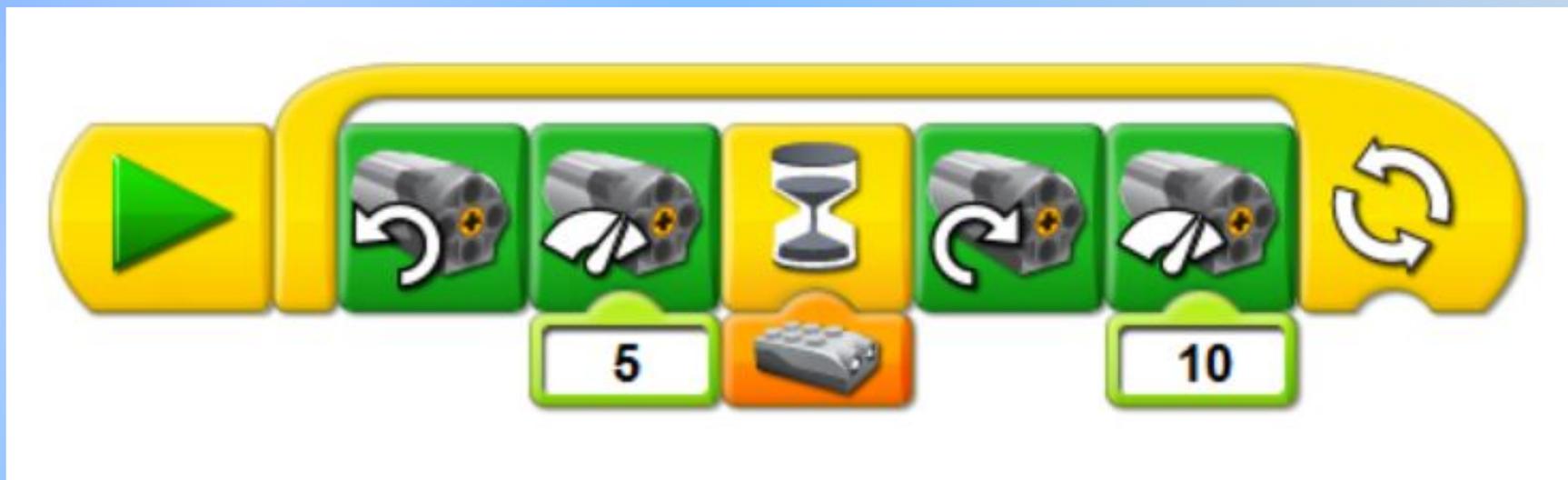
Что будет делать робот при выполнении этой программы?



Что будет делать робот при выполнении этой программы?



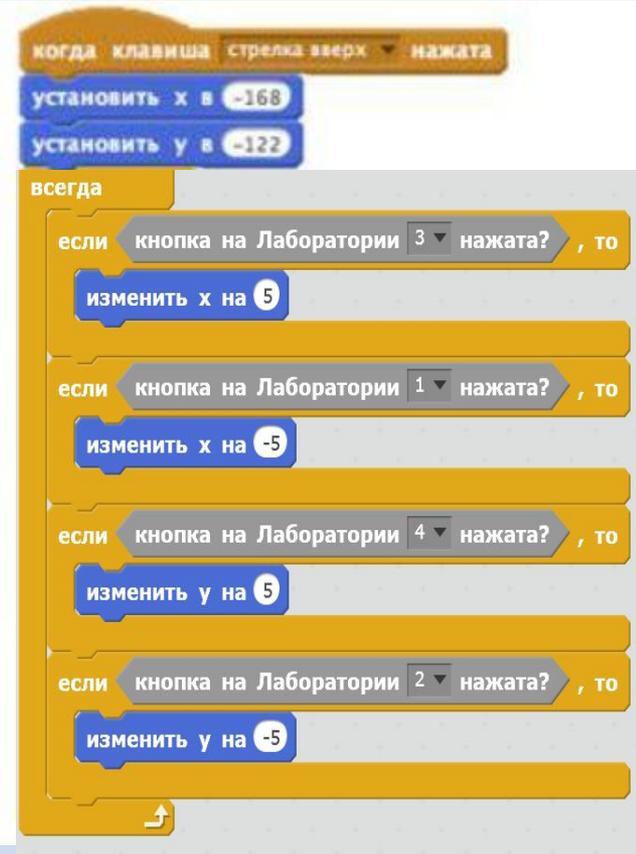
5 этап – освоение ветвящихся алгоритмов «если ... то», «если ... то ... иначе», работа с датчиками



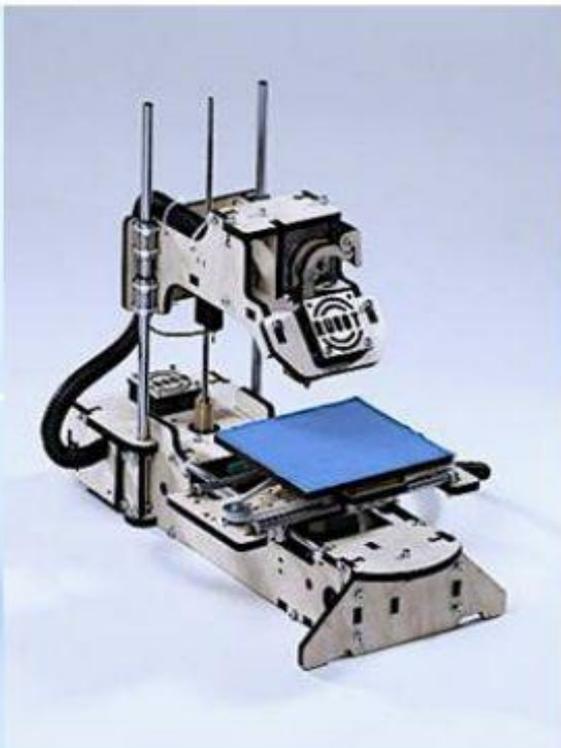
Понаблюдайте, что будет происходить со спрайтом.
Разработайте по аналогии свою игру с другим спрайтом и сценой, поменяйте числовые данные в программе.

ScratchDuino.Лаборатория.

Игра «Послушный кот»



РОББО™



Разработки Санкт-Петербургской компании ROBBO:

- ScratchDuino. Робоплатформа
- ScratchDuino.Лаборатория
- 3D принтер «ROBBO Mini»
- Цифровая лаборатория
- Комплект «Схемотехника»



Чем еще хороши «ScratchDuino»?



- 1 Интуитивно понятны** — облегчают работу педагогам и интересны детям. «СкретчДуино» позволяет объяснить основы электротехники, освоить работу с основными микроэлектронными компонентами, разобраться, как работают датчики, настроить и откалибровать.
- 2 Предназначены для массового внедрения** — в школах, кружках технического творчества, с минимальной подготовкой преподавателя или без нее, для детей от 5 лет.
- 3 Легко подготовить к занятиям и выполнить сложное задание за 10-15 минут** — их можно быстро собрать-разобрать, а затем запрограммировать, учитывая тему урока.
- 4 Выпущены под свободной лицензией** — можно скачать инструкцию из Интернета, изучать, изменять, самим выпускать датчики и запасные части, например на 3D-принтере, и создать устройство «с нуля» — от корпуса до функционала.
- 5 Все компоненты разработаны и производятся в России**, соответствует ФГОС и помогает внедрить единые образовательные стандарты на всей территории Российской Федерации.

Робоплатформа и Лаборатория

ScratchDuino



ROBBO

(Scratch + Arduino)

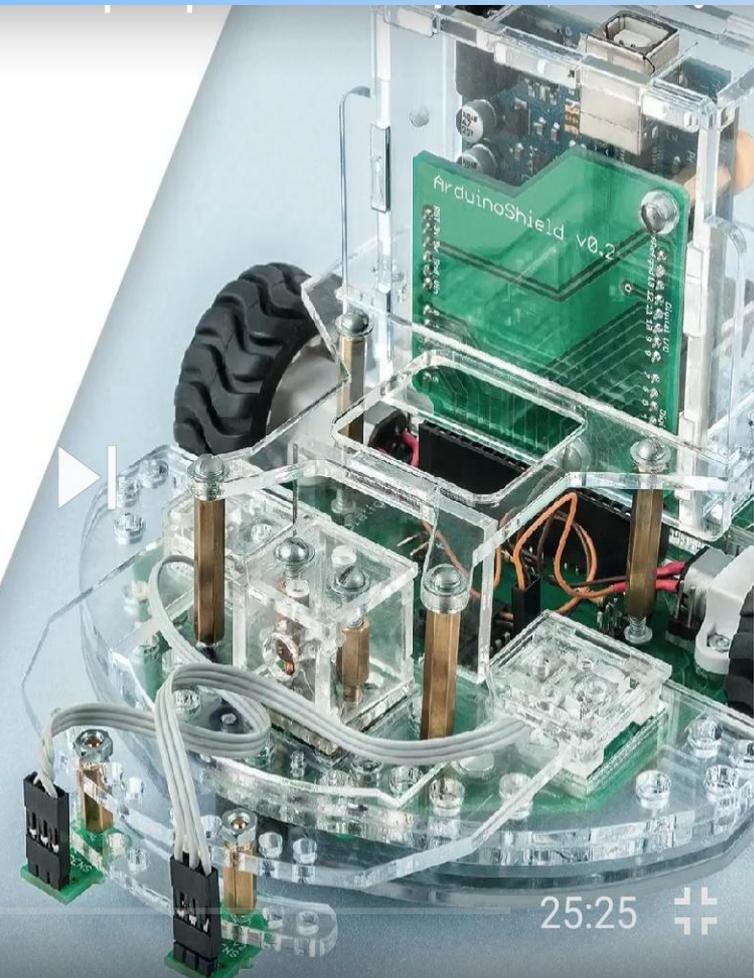


Достаточно уметь читать

ScratchDuino P0550 • 30 просмотров

ROBBO Робоплатформа ScratchDuino

- это модульная система, которую можно менять по своему усмотрению. На ней есть магнитные площадки, чтобы дети могли сами крепить разные датчики: фоторезисторы, датчики касания, линии и любые другие.



12:03

25:25



ScratchDuino ROBBO • 30 просмотров



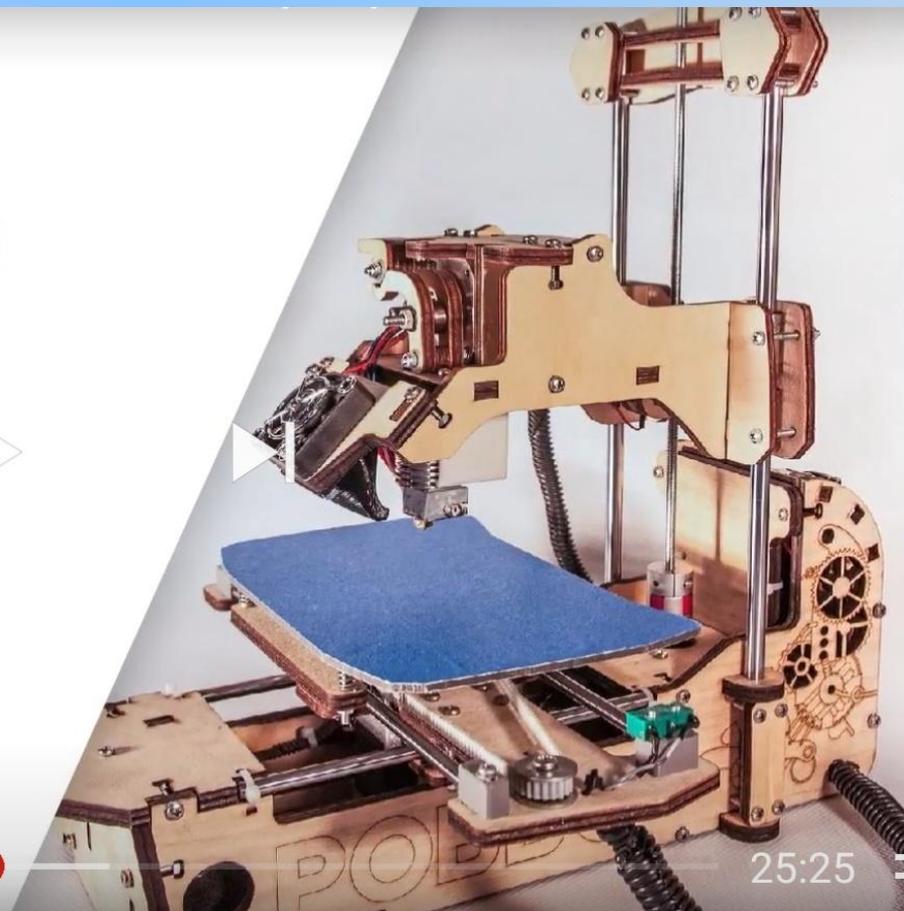
ROBBO Лаборатория ScratchDuino

состоит из платы с датчиками и программируемыми сенсорами, которая упакована в прочный прозрачный картридж, чтобы дети могли видеть содержимое и разобраться в микроэлектронной части.

Scratchduino R0550 • 30 просмотров

ROBBO 3D-принтер МИНИ

- это устройство для создания
трехмерного физического объекта по
созданной на компьютере модели.



13:23

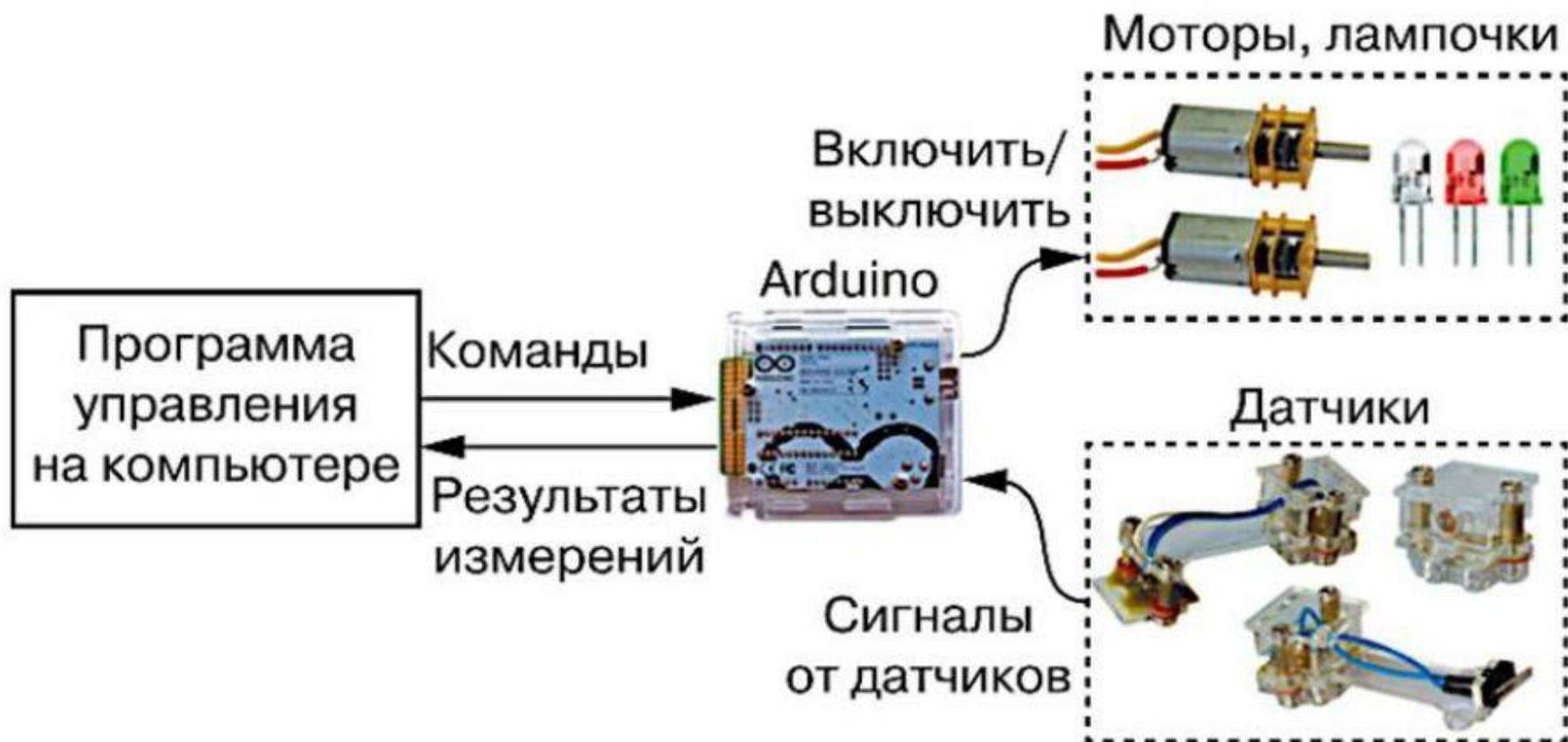
25:25

Робоплатформа

- платформа с колёсами
- микропроцессор Arduino
 - два датчика касания;
 - два датчика освещенности
 - два датчика линии;
 - датчик «инфракрасный глаз»
 - флеш-накопитель (загрузка ОС и необходимого ПО)
 - USB кабель
 - адаптер Bluetooth
 - крепления для батареек типа AA и «Крона»



Как работает робот?



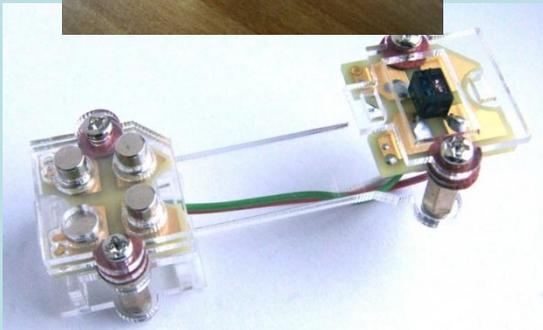
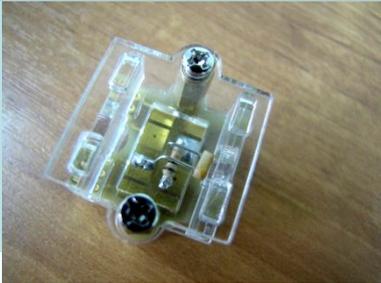
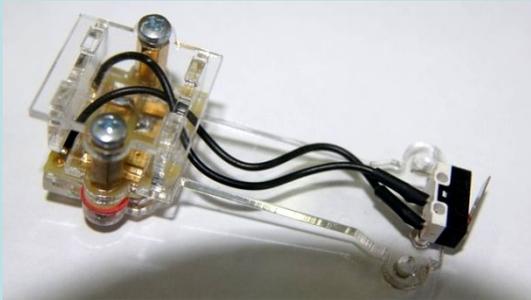
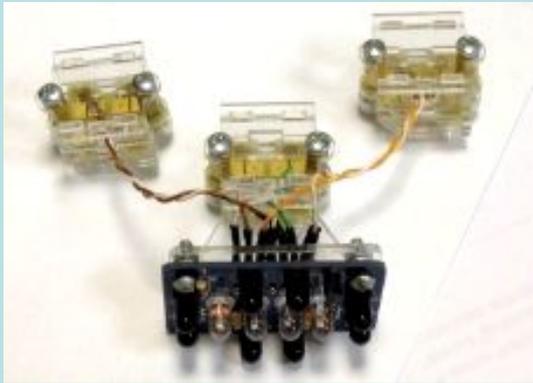
Сенсоры (датчики) ScratchDuino

сигнализирует о столкновении с предметом, может использоваться как кнопка

позволяет ехать по нарисованной линии, обнаруживать край стола

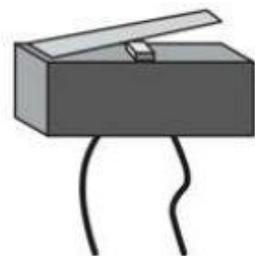
помогает объезжать препятствия, ездить вокруг арены

позволяет двигаться на свет



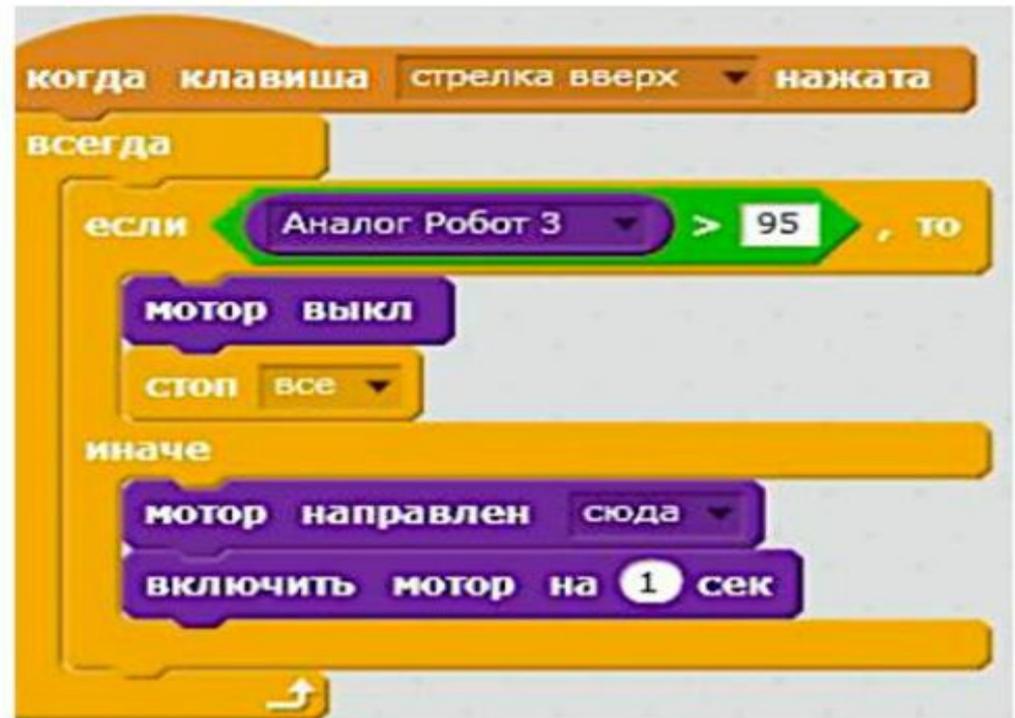
Что такое датчик касания?

Датчик касания — это кнопка, которая замыкает или размыкает контакт. С него может поступить только два различных сигнала — «кнопка не нажата» и «кнопка нажата». Если кнопка не нажата, в компьютер поступает число 0, а если нажата — число 100

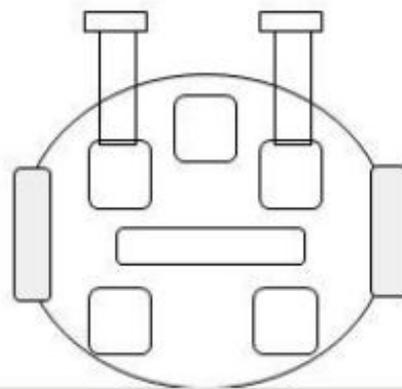


6 этап – считывание показаний датчиков и работа с блоков «операторы» (и, или, >, <, =, +, - и др.)

Датчик касания.



Датчик касания.



Робот датчик 2 ▾

> 80

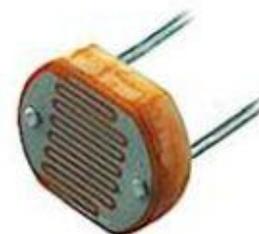
и

Робот датчик 5 ▾

> 80

Что такое датчик освещенности?

Датчик освещенности реагирует на свет, падающий на него. При этом изменяются электрические свойства материала датчика (его электрическое сопротивление) и сигнал на его контактах. Чем больше света получает датчик, тем больше сигнал на его выходе.



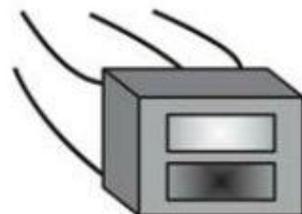
Что такое датчик линии?

Датчик линии состоит из двух элементов: излучателя и приемника инфракрасных лучей. *Излучатель* посылает инфракрасные лучи, а *приемник* их «ловит», уже отраженные поверхностью, на которой стоит робот.

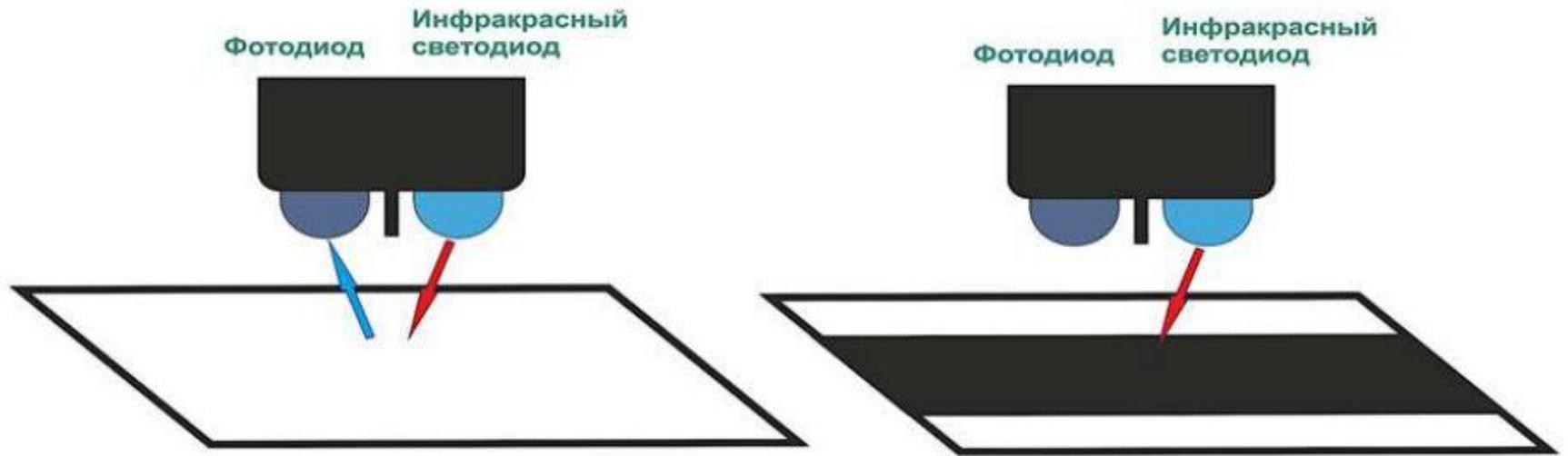
Темная поверхность, поглощает большую часть лучей.

Светлая поверхность отражает почти все лучи.

В ScratchDuino используется особая электрическая схема и получается: на светлой поверхности датчик выдает маленькие значения, а на темной — большие.



Что такое датчик линии?



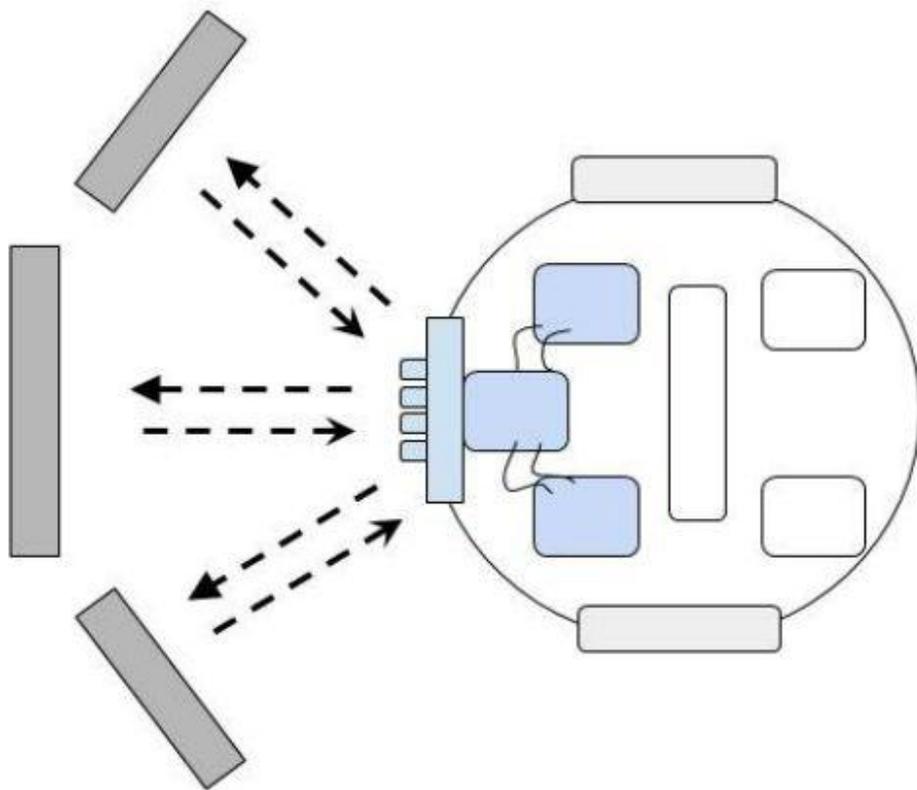
Что такое «инфракрасный глаз»?

«Инфракрасный глаз» (датчик расстояния) — это сложный датчик, который использует инфракрасные лучи.

Он умеет обнаруживать препятствия в трех направлениях: прямо перед собой (в центральной зоне) и в двух боковых зонах (левой и правой). За каждую область отвечают свои передатчики и приемники инфракрасного сигнала.



Что такое «инфракрасный глаз»?



Лаборатория

Базовые сенсоры:

1. Датчик звука (Звук)
 2. Переменный резистор (Рычажок)
 3. Датчик света (Свет)
 4. Разъемы для дополнительных аналоговых датчиков (Сопротивление А и В)
 5. Кнопка
- +
Расширенные сенсоры (программируемые)



Данные от датчиков, расположенных в разных гнездах Робоплатформы отображаются в окошке, каждый под своим номером Аналога. Как понять, какому гнезду какой Аналог соответствует? (можно использовать датчик света или датчик касания, последовательно размещая его во все гнезда платформы и наблюдая за изменением значения 0 в окошке)



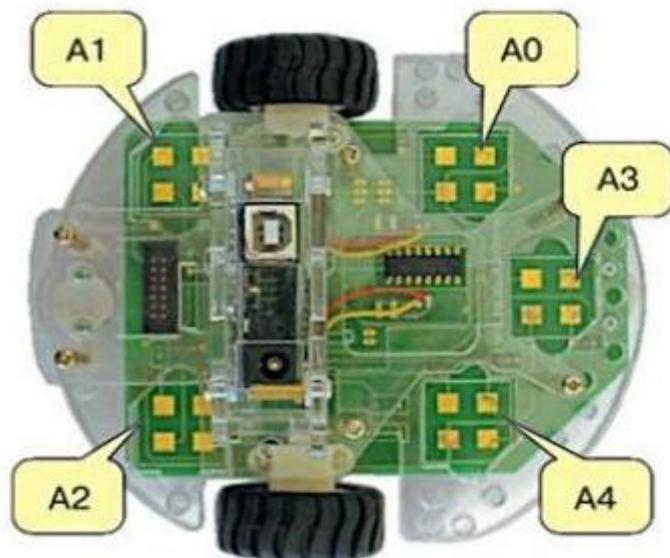
Робот	
Аналог 0:	0
Аналог 1:	0
Аналог 2:	0
Аналог 3:	0
Аналог 4:	0
Кнопка:	false

Нарисуй схему гнезд Робоплатформы (вид сверху) и подпиши каждое гнездо: датчик 1, 2, 3, 4 и 5.

Составь таблицу и запиши минимальные и максимальные значения для каждого из датчиков.

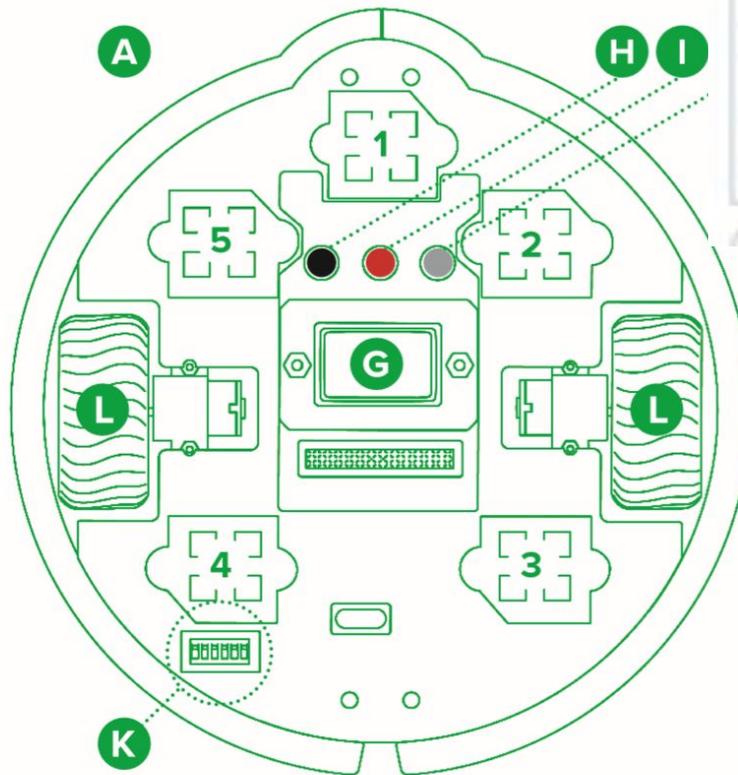
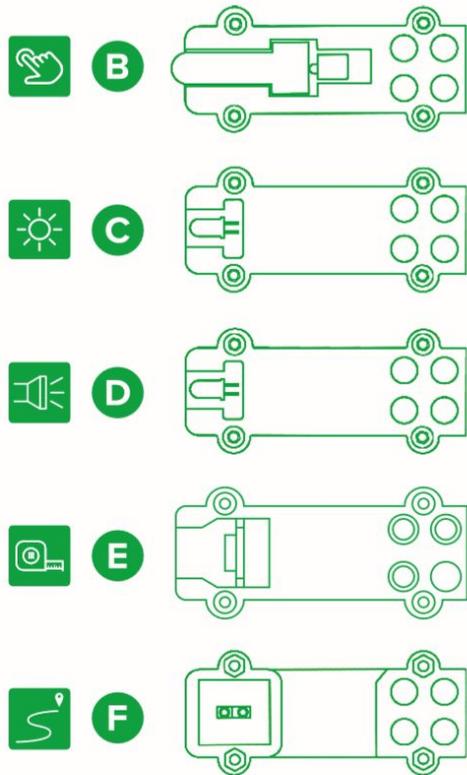
ScratchDuino.Робоплатформа:

- может одновременно вмещать **до 5 датчиков**, которые готовы собирать и обрабатывать данные в соответствии с пользовательской программой
- показания датчиков в режиме **реального времени** отображаются на панели датчиков.



Робот

Аналог 0:	2
Аналог 1:	0
Аналог 2:	0
Аналог 3:	100
Аналог 4:	79
Кнопка:	false



Робот



Сч-к пути Л: --

--

Сч-к пути П: --

--

Датчик 1: --

--



Датчик 2: --

--



Датчик 3: --

--



Датчик 4: --

--



Датчик 5: --

--



Кнопка старт: ложь

Датчик касания.

когда щелкнут по 

если **Робот** датчик 1 > 80 , то

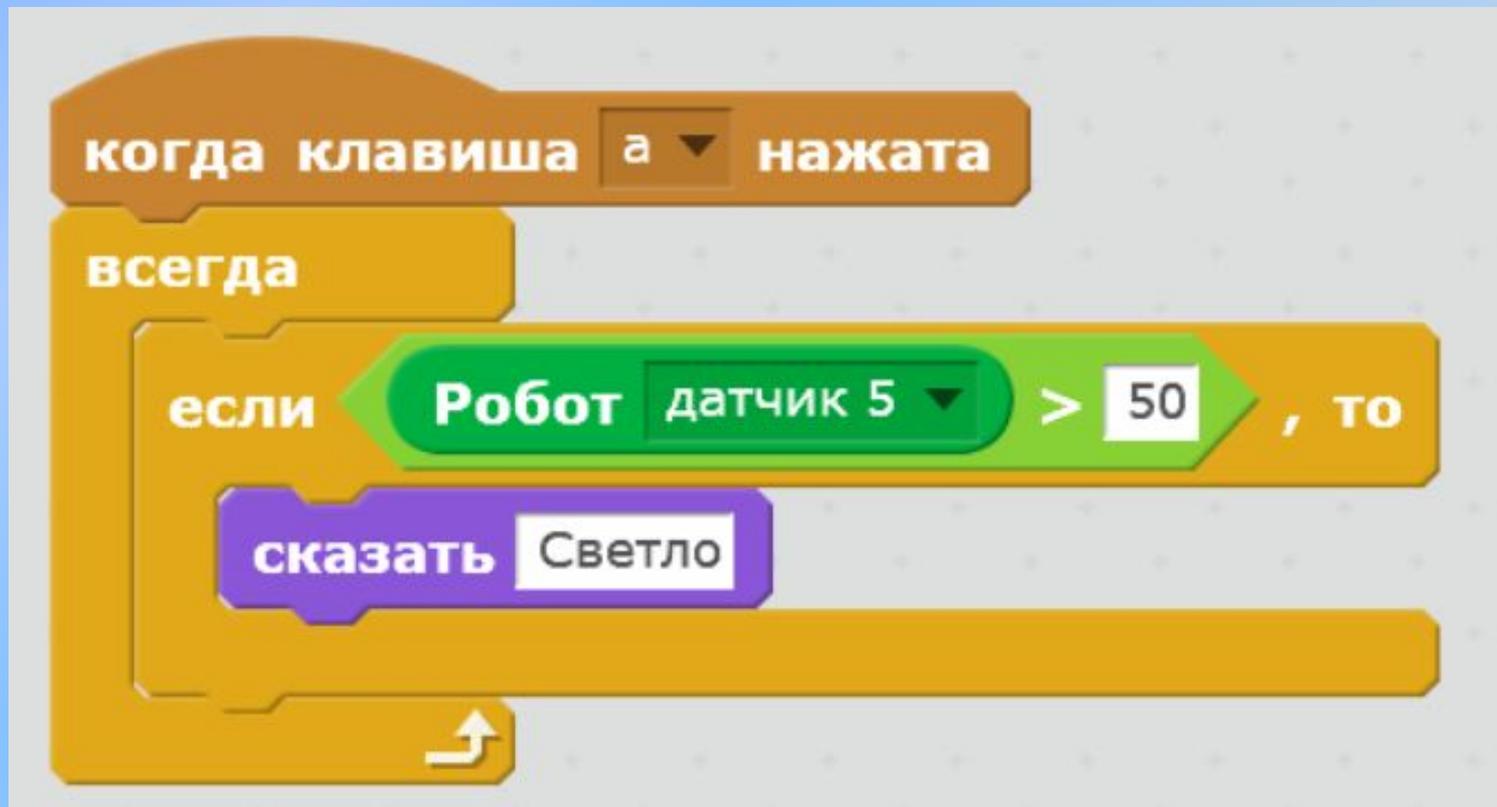
играть звук **роп**

сказать Проход закрыт!

иначе

сказать Путь свободен!

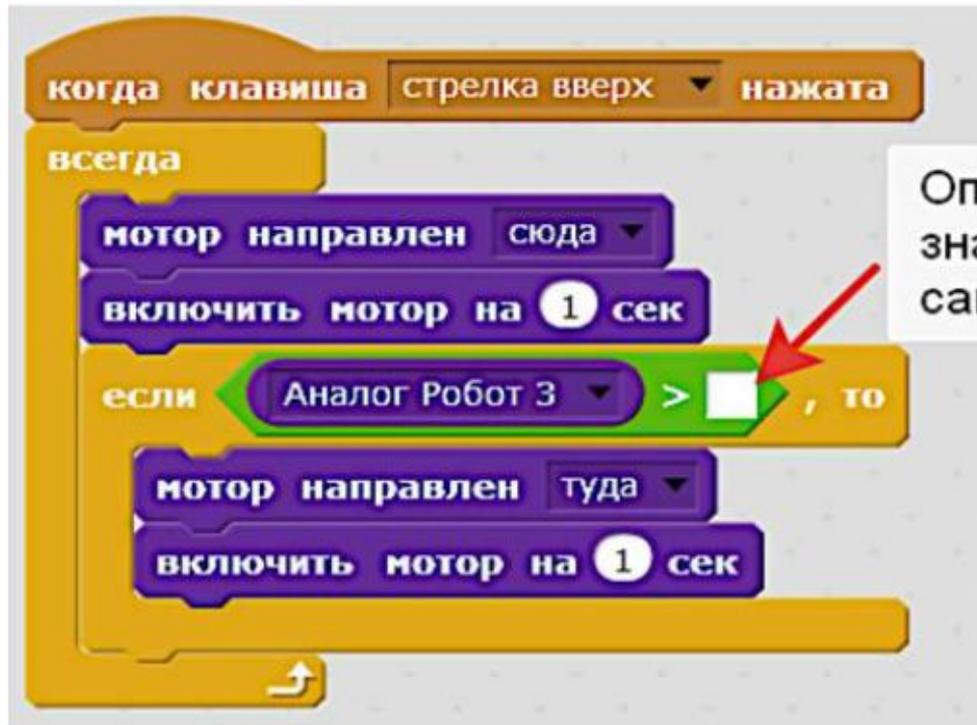
Датчик освещенности.



7 этап – программирование робота по заданным условиям

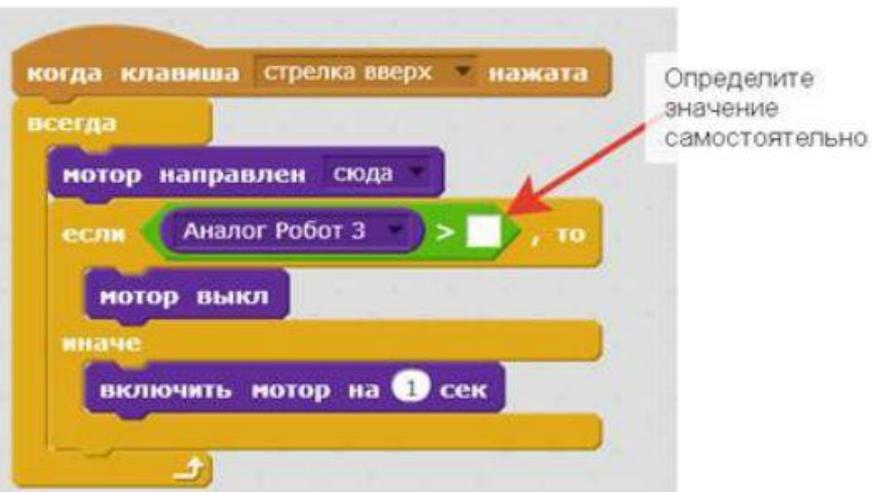
Датчик линии.

Робоплатформа едет вперед по белому листу. Оказавшись на краю листа, она должна отъехать назад



Определите значение самостоятельно

Использование датчика расстояния



Робоплатформа едет вперед.

Если впереди она обнаруживает препятствие, то должна остановиться. Если препятствие исчезает, то Робоплатформа движется дальше.

Создайте программу для Робоплатформы по образцу.

Не забудьте блок-тормоз для экстренной остановки!

Использование СкретчДуино.Лаборатории для управления роботом.



Создайте программу для управления движением Робоплатформы с помощью Лаборатории по образцу.

Если на Лаборатории нажата кнопка “Вверх”, то Робот движется вперед.

Не забудьте блок-тормоз для экстренной остановки!

8 этап – создание проектов с программируемыми роботами

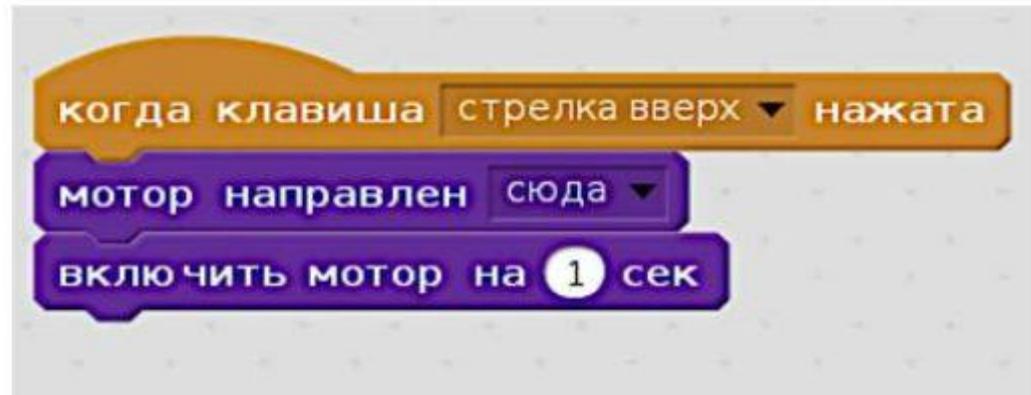
Создание простого проекта

Использование датчиков значительно повышает функциональность робота позволяет проектировать сложные и перспективные проекты.

Создадим программу для робота с **датчиком касания**:

- за основу возьмем программу движения по стрелке
- добавим условие, которое проверяет значение датчика касания
- определим, что должно произойти, если условие выполнено.

Теперь, если робот врежется в какое-то препятствие, он немного отъедет назад.



Типы заданий по программированию:

- 1) создай программу по образцу;
- 2) предположи, как будет двигаться робот;
- 3) сравни алгоритмы, выбери программу из предложенных;
- 4) дополни/продолжи алгоритм;
- 5) измени программу/создай свой алгоритм по заданным условиям;
- 6) проведи исследование, запиши данные в таблицу;
- 7) задай условия для другой подгруппы учащихся;
- 8) создай программу для сконструированного тобой робота / создай собственный анимированный ролик/мультфильм (разработай свой *проект*)

Экспериментирование на занятиях по робототехнике

Для любого занятия по Рбт кроме триединой цели определяются задачи по экспериментированию и программированию.

При использовании LEGO WeDo можно экспериментир. с:

- направлением и скоростью движения модели или её части;
- передачей энергии с помощью как простых (рычаг, шестерня, ремень), так и более сложных механизмов (коронное колесо, кулачок, червяк);
- сменой датчиков;
- изменением поведения модели путём замены деталей или блоков программы (звук, последовательность действий и др.);
- усложнением конструкции.

Экспериментирование на занятиях по робототехнике

При использовании ScratchDuino экспериментируем с:

- управлением роботом (автоматическое исполнение программы или управление каждым движением);
- применением различных датчиков;
- скоростью и маршрутом движения колёсной Робоплатформы (и связанными с этим соревнованиями);
- дополнением базовой конструкции собственными деталями, распечатанными на 3D принтере;
- сменой типов алгоритма в программе;
- измерением температуры и других физических явлений окружающей среды;
- введением переменных (сменой свойств);
- синхронным изменением экранных персонажей (движением, внешностью, звуком спрайтов);
- величинами и математическими операциями и др.

Последовательность экспериментирования

1. Осознание проблемы.
2. Формирование задачи исследования.
3. Продумывание методики эксперимента.
4. Выслушивание инструкций и критических замечаний.
5. Прогнозирование результатов.
6. Выполнение работы.
7. Соблюдение правил безопасности.
8. Наблюдение результатов.
9. Анализ полученных данных.
10. Словесный отчет об увиденном.
11. Формирование выводов



Литература

1. Применение Роббо на уроках в начальной школе [Электронный ресурс] : метод, указания для учителей нач. кл. и студентов направления 44.03.05 "Педагогическое образование" профилями подгот. "Начальное образование", "Начальное образование. Дополнительное образование", "Начальное образование. Иностранный язык" всех форм обучения / В. А. Рожина [и др.] ; ВятГУ, Педагогический институт, ФПП, каф. ПМДНО. - Киров : [б. и.], 2017. - 87 с.
2. Бочкала, А. Е. Общедоступное программирование в Scratch / А. Е. Бочкала. Режим доступа: <http://scratch.uvk6.info>
3. Винницкий, Ю.А. Конструируем роботов на ScratchDuino. Первые шаги. — М. : Лаборатория знаний, 2016. – 116 с.

Литература

4. Вострикова, Е. А. ScratchDuino.Робоплатформа : руководство пользователя / Е. А. Вострикова, Л. С. Захаров, Е. А. Львова. — Санкт-Петербург : Множительный центр ЗАО «Тырнет», 2015. — 70 с.
5. Вострикова, Е. А. ScratchDuino.Лаборатория : руководство пользователя / Е. А. Вострикова, Л. С. Захаров, Е. А. Львова. — Санкт-Петербург : Множительный центр ЗАО «Тырнет», 2015. — 54 с.
6. Пашковская, Ю. В. Творческие задания в среде Scratch. Рабочая тетрадь для 5-6 классов. — М. : БИНОМ. Лаборатория знаний, 2014. — 200 с.
7. РОББО. Самоучитель для преподавателей и родителей. СПб., [б.и.] – 110 с.
8. Скретч Вики. Режим доступа:
Scratch-ru.info/wiki/Заглавная_страница

A decorative graphic on the left side of the slide consists of several thick, curved arrows in various colors (purple, red, blue, yellow, green) pointing upwards and to the right, overlapping each other.

Благодарю за внимание!

*Рожина В.А.,
доцент каф. ПМДНО, к.пед.н.*