



В. В. Тарапата, Б. В. Прокофьев

Учимся вместе со **SCRATCH**



когда щелкнут по
здать а значение 0
всегда
спросить Да или нет? и ждать
если ответ = Да, то
здать а значение
сказать Запомнил
сменить фон

Программирование,
игры, робототехника



В. В. Тарапата, Б. В. Прокофьев

Учимся вместе со SCRATCH

Программирование,
игры, робототехника



Москва
Лаборатория знаний

УДК 087.5:004.9
ББК 32.67
Т19

Серия основана в 2018 г.

Тарапата В. В.

Т19 Учимся вместе со Scratch. Программирование, игры, робототехника / В. В. Тарапата, Б. В. Прокофьев. — М. : Лаборатория знаний, 2019. — 228 с. : ил. — (Школа юного программиста).

ISBN 978-5-00101-177-4

До настоящего программиста нужно дорасти, а в школе серьезному программированию начинают учить только в 8–9 классах — теперь это в прошлом! «Учимся вместе со Scratch» — новый трамплин в мир программирования для всех ребят в возрасте 9–12 лет. Визуальная среда Scratch позволяет изучать основы программирования в увлекательной и интуитивно понятной форме, а благодаря тщательно продуманной методике она становится мощным инструментом, который позволит подготовиться к изучению профессиональных языков программирования и достижению самых высоких результатов! Обучение разделено на два этапа: в первом мы изучим классическое программирование в той же последовательности изложения, в какой изучают любой другой серьезный язык программирования. На втором этапе мы займемся разработкой трех видеоигр и даже подключим геймпад и запрограммируем управление от него (требуется ScratchDuino.Лаборатория или ROBBO LAB). Каждая глава пособия — уровень, достигая каждый из которых мы движемся к званию настоящего программиста.

УДК 087.5:004.9
ББК 32.67

6+

Издание для дополнительного образования

Серия: «Школа юного программиста»

Тарапата Виктор Викторович

Прокофьев Борис Викторович

**УЧИМСЯ ВМЕСТЕ СО SCRATCH.
ПРОГРАММИРОВАНИЕ, ИГРЫ, РОБОТОТЕХНИКА**

Для детей среднего и старшего школьного возраста

Ведущий редактор *Ю. А. Серова*. Художественный редактор *В. А. Прокудин*

Технический редактор *Т. Ю. Федорова*. Корректор *И. Н. Панкова*

Компьютерная верстка: *Е. Г. Ивлева*

Подписано в печать 18.09.18. Формат 70×100/16.

Усл. печ. л. 18,85. Заказ

Издательство «Лаборатория знаний»

125167, Москва, проезд Аэропорта, д. 3

Телефон: (499) 157-5272

e-mail: info@pilotLZ.ru, <http://www.pilotLZ.ru>

Содержание

От авторов	3
Благодарности	4
Введение	5
Загрузка и установка Scratch 2.0 Offline Editor	9
РАЗДЕЛ 1. Программирование	11
Обзор интерфейса	12
Первая программа	17
Циклы	27
Декартова система координат	36
Рисование по координатам	48
Псевдослучайные числа, интерактивность, управление программами	63
Переменные	85
Ввод данных	100
Математический тренажер	113
РАЗДЕЛ 2. Создание игр	119
ROBBO Lab	120
Arkanoid	123
«Змейка»	152
Pong	184
Сообщество Scratch	221
Заключение	223
Словарь программиста	224
Приложение. Математический тренажер	227

От авторов

Дорогие друзья!

У вас в руках книга, которая откроет перед вами совершенно новые горизонты. Вас ждет волшебный мир программирования и разработки увлекательных компьютерных игр!

Главный ваш спутник в этом путешествии — среда программирования Scratch. В ней можно создавать самые разные интерактивные программы с анимацией, а также мультфильмы и игры.

Книга состоит из двух разделов: «Программирование» и «Создание игр».

В первом разделе мы с вами освоим основы программирования, пройдя уровни от начинающего до настоящего программиста. Одновременно будем создавать несложные анимации и мини-игры.

Во втором разделе мы применим полученные знания для создания трех действительно серьезных игр. Чтобы почувствовать себя настоящими разработчиками, для каждой из них вам предстоит:

- ✓ создать первоначальную версию игры **1.0**;
- ✓ произвести *апгрейд* игры до версии **2.0**;
- ✓ подключить *геймпад* и запрограммировать его для управления игрой¹;
- ✓ попробовать расширить функционал игры для очередного апгрейда до версии **3.0**.

Надеемся, что наша книга станет для вас путеводной звездой в мире программирования и разработки игр, позволит реализовать ваши самые смелые творческие идеи и добиться успехов в учебной и будущей профессиональной жизни!

Удачи вам, друзья! Дерзайте!

¹ Данный пункт не является обязательным, если вы не располагаете ROBBO Lab или ScratchDuino.Лаборатория. Все игры создаются изначально для управления с помощью клавиатуры и мыши.

Благодарности

Авторы выражают безграничную благодарность всем людям, принявшим участие в создании, редактировании и совершенствовании этой книги, а также тем, кто, так или иначе, повлиял на ее выход в свет.

Отдельно авторы благодарят:

- своего бессменного научного руководителя, кандидата педагогических наук, профессора МПГУ Надежду Николаевну Самылкину за неоценимый вклад в образование и помощь при написании книги;
- учащихся школы № 283 города Москвы за вдохновение, огромный интерес, помощь в бета-тестировании и написании книги, а именно всех учащихся 6 «Д» и 6 «Е» классов, в частности Александра Сенина, Никиту Шмурова, Василису Шумилину, Нику Шумилину, Рано Шарипову, Тамару Меджидову, Анну Чебутаеву, Дмитрия Семенякина, Ярослава Шикова, а также Дмитрия Шикова — ученика 7 «Е» класса, Снежану Дудкину, Елизавету Ляднову — учениц 9 «Д» класса, Софию Салову, Глеба Кордабовского, Андрея Тарасенко — учеников 10 «А» класса;
- Михаила Николаевича Бородина, Ольгу Васильевну Карцеву, Сергея Вячеславовича Гончаренко, Александра Александровича Елизарова, Андрея Красных, Алену Салахову — наших наставников и уважаемых коллег — за конструктивную критику и ценные советы;
- Павла Фролова, продюсера проекта ROVBO, за профессиональное содействие и помощь;
- Веру Андреевну Словиковскую, учителя информатики Лицея № 1568 города Москвы, за созданный интерес к информатике и программированию в школьные годы;
- Лилию Анатольевну Тарапата, Нину Алексеевну Прокофьеву — наших дорогих и любимых мам, которые всегда были рядом и верили в нас.

Введение

Алгоритмизация и программирование

Персональный компьютер, ноутбук, планшет, смартфон и прочие гаджеты на самом деле не больше, чем набор деталей, «упакованных» в корпус. Этот набор еще принято называть **hardware**¹.

Программисты вдохнули жизнь в грудку пластика и металла, создав для их «тяжелого»² содержимого программное обеспечение, называемое **software**³.

Дело в том, что все перечисленные выше устройства, которыми мы пользуемся ежедневно, содержат процессор — электронный «мозг», который обрабатывает информацию и, по сути, позволяет компьютеру быть «умным» устройством. Процессор понимает двоичный код (0 — электричества нет, 1 — электричество есть), в который превращаются все команды алгоритма, написанные на любом языке программирования.

Чтобы компьютер мог выполнять сложные задания, для процессора нужно составить **алгоритм** — строгую и предельно логичную последовательность действий для решения конкретной задачи. Процесс создания алгоритма и называется **алгоритмизацией**.

И вот какие требования традиционно предъявляются к любому алгоритму:

- 1) однозначность — все шаги алгоритма должны строго следовать друг за другом, иметь однозначную трактовку и не допускать никаких двусмысленностей;
- 2) конечность — алгоритм не может исполняться бесконечно; после того как задача решена или стало понятно, что она не может быть завершена, алгоритм обязан прерваться;

¹ В переводе с английского — аппаратное обеспечение, компьютерные комплектующие, «железо».

² *Hard* (часть слова *hardware*) переводится с английского как «тяжелый».

³ В переводе с английского — программное обеспечение, комплекс программных средств.

- 3) **детерминированность** — алгоритм должен работать одинаково при одинаковых начальных условиях и выдавать каждый раз один и тот же результат;
- 4) **корректность** — результат работы алгоритма должен соответствовать начальным условиям;
- 5) **массовость** — один и тот же алгоритм должен подходить для решения класса однотипных задач;
- 6) **эффективность** — ресурсы, затрачиваемые на исполнение алгоритма, должны быть оптимально распределены (для компьютера такими ресурсами являются память и время);
- 7) **понятность** — алгоритм должен быть понятен тому, кто его исполняет.

В этом списке седьмой пункт и порождает понятие **программы** — алгоритма, переведенного на язык, понятный компьютеру. Именно тот алгоритм, который мы составляем для решения конкретной задачи, создается на том естественном языке, который мы знаем (русский, английский и др.). Компьютер, в свою очередь, понимает всего лишь два цифровых сигнала: 0 и 1.

Вопрос: как заставить процессор компьютера выполнять нужные команды?

Ответ: нужно обратиться к человеку, который способен перевести алгоритм на язык, понятный компьютеру. Такого человека называют **программистом**, а процесс перевода алгоритма на понятную процессору последовательность действий — **программированием**.

Заметим, что мы представили программиста как переводчика, способного создать прямой диалог между человеком и компьютером. И это вовсе не зря, ведь для «общения» с компьютером программист использует **язык программирования**. Одним из таких языков и является Scratch.

Что такое Scratch?

Scratch — это визуальная среда разработки программ, созданная специально для обучения основам програм-

мирования. Она сочетает самые лучшие решения для начинающих:

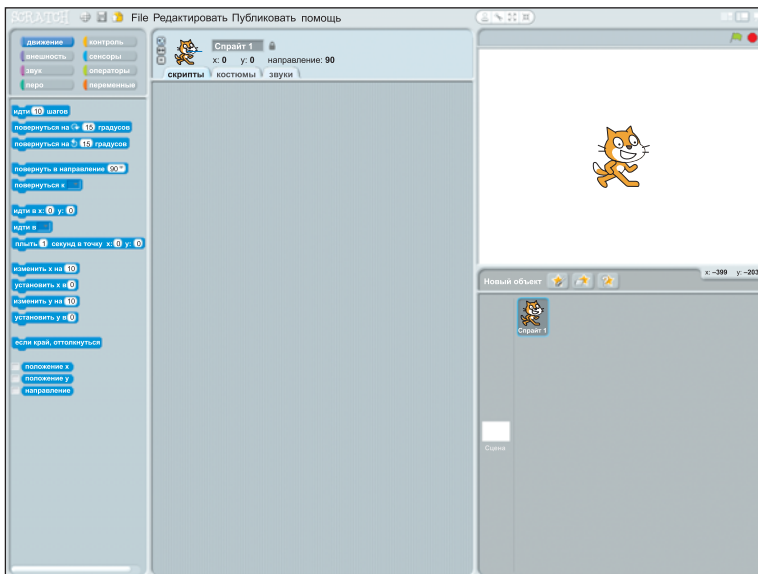
- все основные алгоритмические структуры и конструкции, используемые в программировании;
- визуальные блоки, заменяющие текстовые команды;
- анимационное поле, объекты на котором подчиняются создаваемой из блоков программе;
- встроенные инструменты для создания и редактирования творческих решений;
- свободное (бесплатное) распространение.

Создавая анимацию, игры и интерактивные модели в Scratch, на самом деле мы учимся именно программировать и готовимся к покорению новых вершин.

Среда Scratch создана группой ученых и разработчиков Массачусетского технологического университета¹, США.

На сегодняшний момент используются две версии Scratch: 1.4 и 2.0.

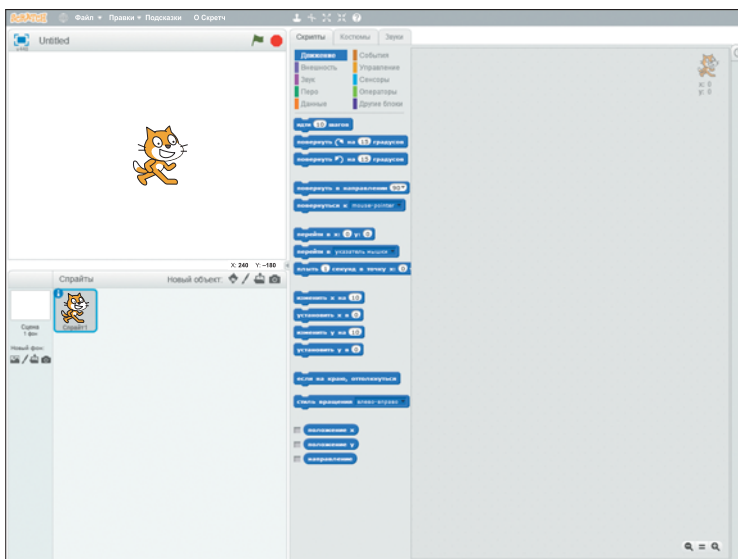
Scratch 1.4 — первая рабочая версия, разработанная на основе языка Squeak.



Scratch 1.4

¹ MIT — Massachusetts Institute of Technology, USA.

Scratch 2.0 — вторая версия, адаптированная для работы онлайн и переписанная на языке Flash, использующем широко распространенную технологию работы приложений в сети Интернет.

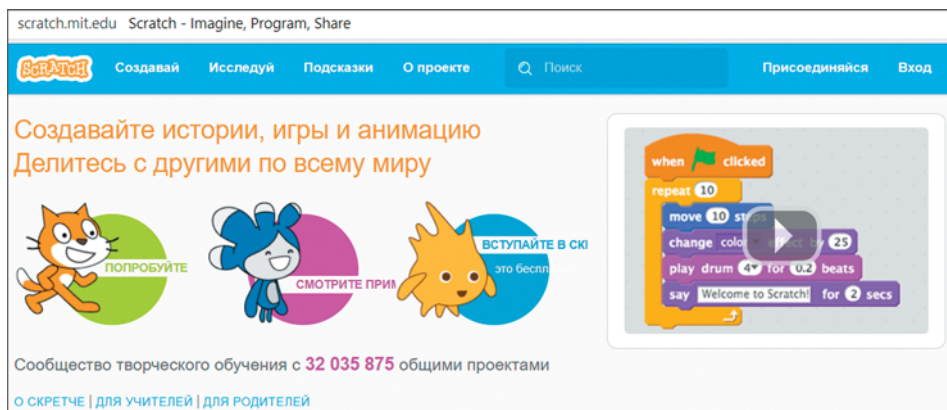


Scratch 2.0

Мы будем работать с версией Scratch 2.0, так как именно она является самой новой.

Scratch доступна на официальном сайте разработчика для работы в онлайн-режиме: <https://scratch.mit.edu/>.

Для начала работы с ней в верхнем меню нужно нажать кнопку **Создавай**.

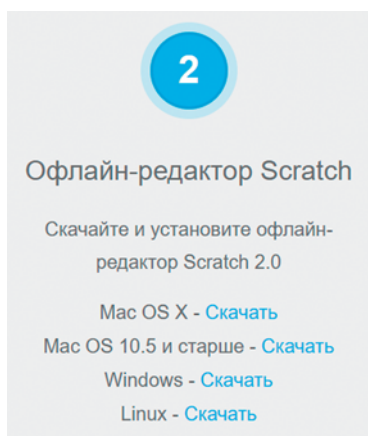


Загрузка и установка Scratch 2.0 Offline Editor

Помимо работы онлайн Scratch можно скачать для работы без подключения к сети Интернет. Так работать намного надежнее, потому что мы не будем зависеть от стабильности работы Интернета.

Такая версия среды Scratch называется офлайн-редактор Scratch 2.0 и доступна для свободного скачивания с официального сайта разработчика по ссылке: <https://scratch.mit.edu/scratch2download/>. Также этот раздел можно найти в нижней части главной страницы сайта Scratch под названием «Несетевой редактор».

При переходе по ссылке мы попадаем на страничку загрузки Scratch 2.0 Offline Editor. Обратим внимание только на второй шаг:



Здесь необходимо выбрать операционную систему, установленную на компьютере или ноутбуке, и соответственно кликнуть по нужной надписи **Скачать**.

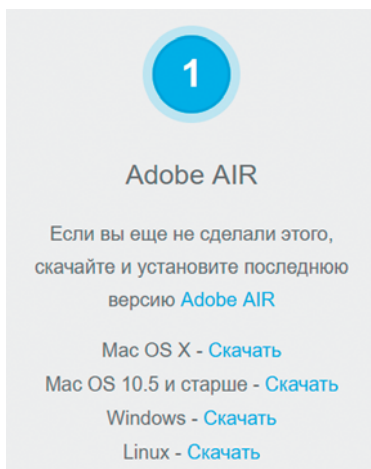
Начнется скачивание установочного файла, после чего его нужно запустить и следовать инструкциям по установке.

Внимание!

Во время установки может понадобиться режим Администратора компьютера (или пароль для него).

Внимание!

Во время установки может возникнуть ошибка из-за отсутствия надстройки Adobe AIR, необходимой для работы Scratch. Мастер установки предложит загрузить ее автоматически — нужно ответить согласием. Если этого не произошло, но ошибка сохраняется, следует снова перейти на страничку загрузки Scratch 2.0 Offline Editor и скачать эту надстройку на шаге 1.

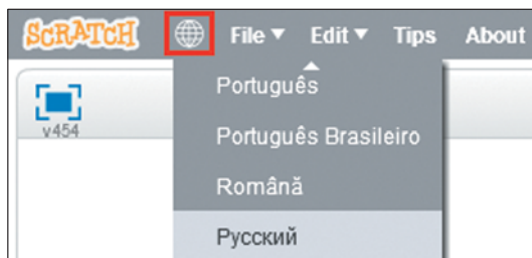


РАЗДЕЛ 1. Программирование

Первый раздел книги посвящен непосредственно программированию. В нем мы вместе пройдем девять уровней — от нулевого до восьмого — на пути к званию «программист».

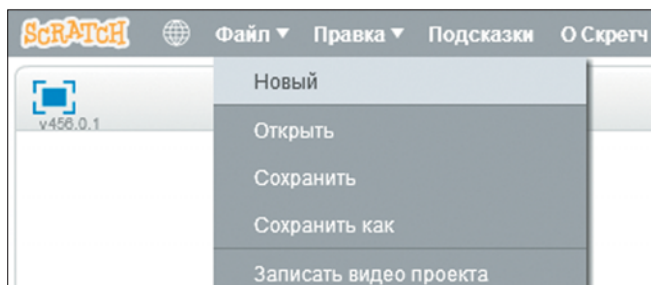
Структура каждого уровня, кроме нулевого, довольно проста: сначала мы вместе пошагово изучаем те или иные приемы программирования, создавая несложные анимации и мини-игры, а затем, для прохождения уровня и получения нового звания, вы самостоятельно выполняете предложенные задания.

При первом запуске Scratch по умолчанию установлен английский язык. Сменить его на русский (или любой другой из доступных) можно, кликнув на значок глобуса в верхней панели меню и инструментов.

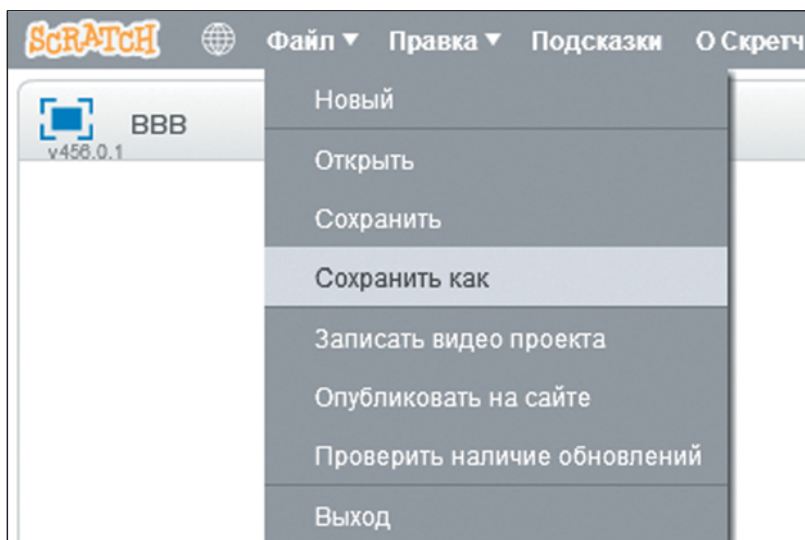


Рекомендуем для каждого нового уровня создавать отдельный проект (программу) в среде Scratch и сохранять свои наработки.

Для создания нового проекта нужно воспользоваться пунктом меню **Файл** → **Новый** (или просто запустить Scratch заново).



Для сохранения своего проекта во избежание его внезапной потери (например, из-за перепада электроэнергии) и его дальнейшей демонстрации друзьям или в сети Интернет нужно воспользоваться пунктом меню **Файл** → **Сохранить как**, затем выбрать папку для его хранения и задать имя.



Также рекомендуем почаще пользоваться пунктом меню **Файл** → **Сохранить** для сохранения изменений в уже созданном проекте.

Вперед, на аллею славы великих программистов!

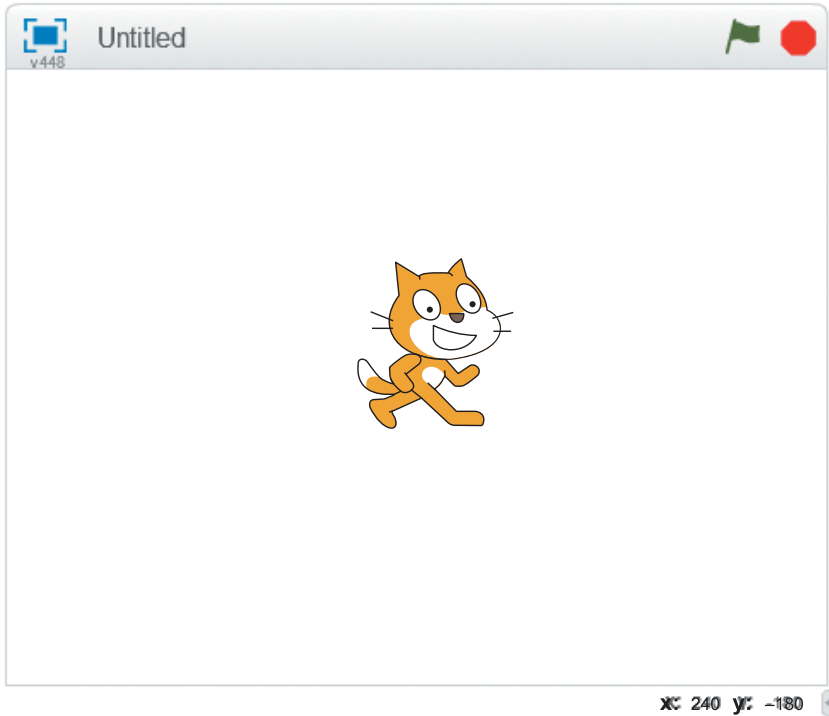
Обзор интерфейса

УРОВЕНЬ 0: «Джуниор»¹

Разберемся, из каких областей состоит рабочее пространство Scratch 2.0, и договоримся о том, как будем называть те или иные элементы.

¹ Джуниор (от англ. *Junior* — младший) — так называют молодых программистов с небольшим опытом или вовсе без него.

Область анимации



В левой верхней части расположена область анимации, на которой мы видим главного героя кота Царапку (Scratch в переводе с английского означает «царапать», поэтому логично предположить, что кота зовут именно так).

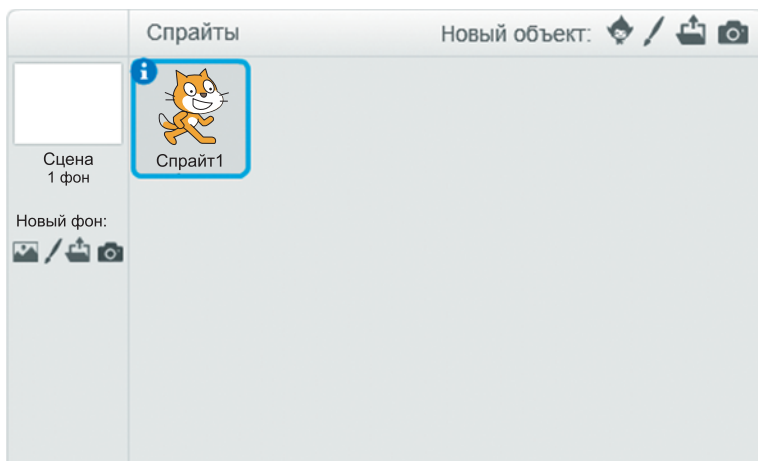
В области анимации мы будем видеть результат работы создаваемых нами программ.

Синий значок в левом верхнем углу позволяет раскрывать область анимации во весь экран.

Зеленый флажок и **красный восьмиугольник** в правом верхнем углу — это кнопки запуска и остановки исполнения программ соответственно.

Также на верхней панели отображается **имя** текущего проекта (по умолчанию Untitled), а на нижней панели — положение указателя мыши относительно активного объекта в координатах x и y .

Область управления объектами



В этой области мы можем выбирать (активировать) объекты для работы с ними и создания программ для них. В новом (пустом) проекте изначально мы можем управлять **Царапкой** и **Сценой**.

Все объекты, с которыми мы работаем в Scratch, еще называются **спрайтами**.

В правом верхнем углу (слева направо) находятся инструменты для:

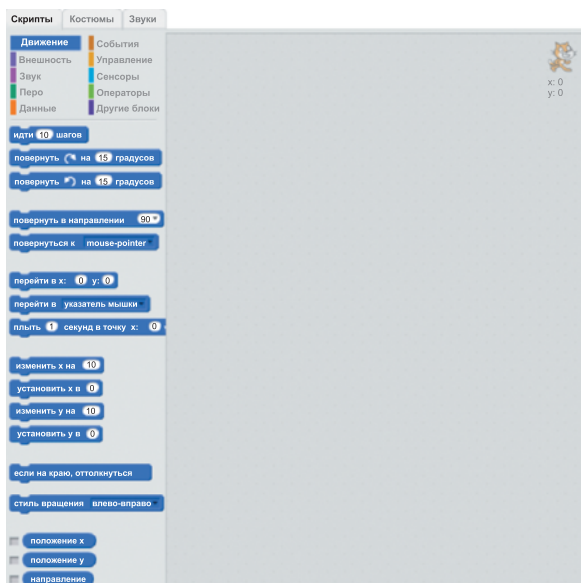
- 1) добавления нового спрайта из стандартной библиотеки;
- 2) создания собственного спрайта с нуля с помощью встроенного графического редактора;
- 3) загрузки спрайта из файла;
- 4) создания нового спрайта с помощью камеры.

Слева — инструменты для работы со сценой и ее фонами.

Следующие три области, которые мы рассмотрим, открываются по вкладкам.

Область скриптов

Это самая главная область. Здесь происходит написание программ. В Scratch они также называются **скриптами**.



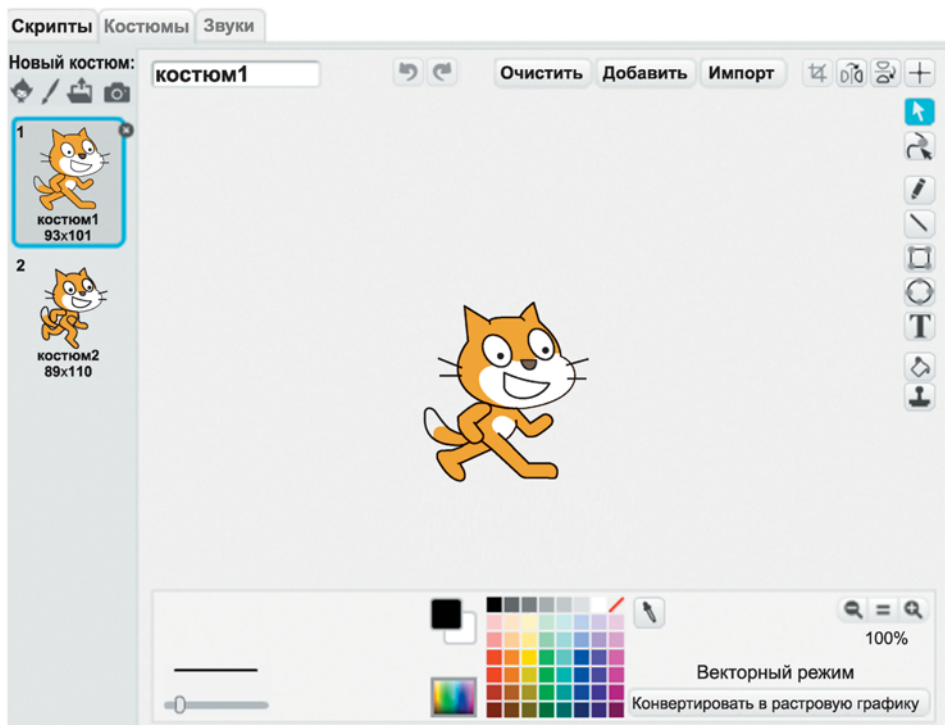
В левой верхней части мы видим группы команд:

Группа	Описание
Движение	Управление положением и перемещением спрайта по области анимации
Внешность	Изменение внешности спрайта (костюмы, размеры, эффекты) и вывод на экран текстовых фраз
Звук	Работа со звуками, издаваемыми спрайтами
Перо	Рисование различных объектов в области анимации с помощью находящегося там спрайта
Данные	Создание и управление переменными и массивами данных (списками)
События	Команды управления событиями, происходящими со спрайтами
Управление	Алгоритмические конструкции проверки условий, циклов, задержки и работа с клонами спрайтов
Сенсоры	Стандартные команды проверки и обработки событий и входных данных
Операторы	Арифметические и логические операции, работа с псевдослучайными числами, обработка строк
Другие блоки	Резервная группа для создания собственных команд и подпрограмм

Скрипты состояются из различных **команд**. В Scratch команды еще называются **блоками** и помещаются программистом в правой части данной области, цепляясь друг за друга, как пазлы.

Также в правой части можно видеть положение спрайта относительно центра сцены в координатах x , y .

Область костюмов

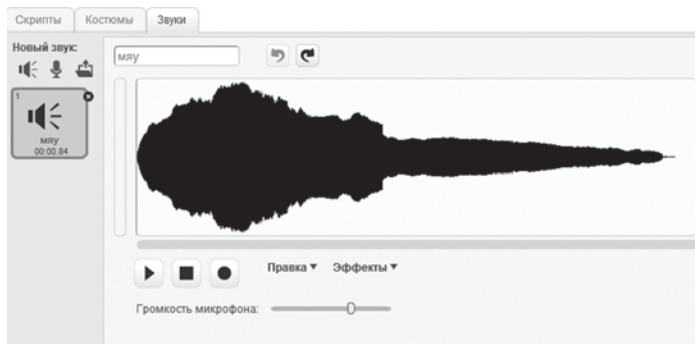


В этой области находятся все доступные костюмы (виды) активного спрайта.

Здесь их можно редактировать как в векторном режиме (работа с графическими примитивами, высокое качество), так и в растровом (точечном) с помощью вполне стандартных инструментов, которые можно встретить в любом графическом редакторе.

Также для костюмов доступно редактирование их имен.

Область звуков



В этой области для активного спрайта можно добавлять звуки, которые он воспроизводит, и редактировать их.

Область меню и инструментов



В верхней части окна находится панель меню и инструментов для работы со всем проектом.

Первая программа

УРОВЕНЬ 1: «Начинающий программист»

Создадим нашу первую программу, чтобы ближе познакомиться со средой Scratch.

Перед тем как приступить к написанию скрипта, определимся с теми действиями, которые должен будет выполнить Царапка, то есть опишем логику программы.

Логика программы

Царапка будет идти вправо несколько шагов, затем влево, после чего скажет: «Привет, Мир!».

Первый шаг

Для начала попробуем сделать так, чтобы Царапка шел вправо.



Перетащим блок **Идти ... шагов** из группы **Движение** в рабочую область скриптов.

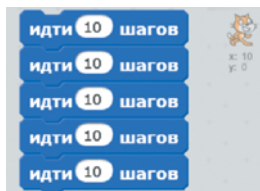
Пока ничего не происходит, потому что команда не приведена в исполнение. Сейчас мы можем проверить, работает ли она. Для этого кликнем по ней один раз.

Мы можем заметить, что Царапка передвинулся немного вправо в области анимации, а его положение относительно центра сцены изменилось на **10** по оси **x**.



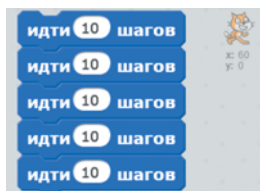
Отсюда мы делаем вывод, что шаги в этом блоке — на самом деле количество единиц (пикселей на экране), на которые смещается спрайт.

Давайте добавим еще четыре такие команды, чтобы Царапка смог прошагать дальше.



Для выполнения этого скрипта, состоящего уже из пяти блоков, кликнем по верхнему из них один раз.

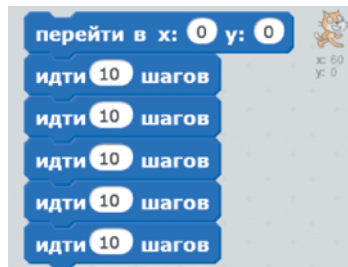
Мы можем заметить, что Царапка сместился намного дальше вправо, чем в предыдущий раз, а его положение изменилось на **50** по оси **x**.



Но мы ведь хотим, чтобы Царапка шел, а не «дергался» вправо. Для этого мы должны добавить несложную анимацию.

Сначала сделаем так, чтобы в начале программы кот возвращался в середину сцены, то есть в точку с координатами **х: 0, у: 0**.

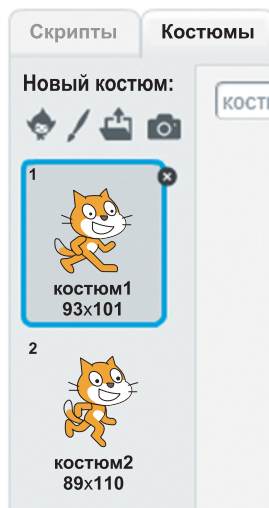
Для этого в начало скрипта добавим команду **Перейти в х: ... у: ...**.



Значения в окошках оставим по умолчанию — именно они нам и нужны.

Теперь перейдем к анимации. Что такое анимация по своей сути? Это смена различных кадров с определенной скоростью.

Для анимации шагов Царапки будем менять два его стандартных костюма, которые можно увидеть на вкладке **Костюмы**.



[. . .]



Программирование – это грамотность XXI века!

Книги новой серии «Школа юного программиста» издательства «Лаборатория знаний» построены на методике пошагового обучения программированию. Следуя этой методике, любой желающий, от школьника до студента вуза, сможет научиться писать программы, разрабатывать мобильные приложения и компьютерные игры и даже освоить технологии машинного обучения и нейросетей.

В серию войдут следующие учебные пособия:

- «Учимся вместе со Scratch: программирование, игры, робототехника» (5–6 классы)
- «Scratch 2.0: от новичка к продвинутому пользователю. Пособие для подготовки к Scratch-Олимпиаде» (1–11 классы)
- «Творческие задания в среде Scratch. Рабочая тетрадь для 5–6 классов»
- «Scratch 2.0: творческие работы на вырост. Рабочая тетрадь для 7–8 классов»
- «Создаем игры с Kodu Game Lab» (4–5 классы)
- «Python для начинающих – от основ до ООП и приложений» (7 класс)
- «Олимпиадное программирование на Python» (7–8 классы)
- «C# – новый учебный курс программирования от основ до продвинутого уровня» (8–9 классы)
- «Android-разработка: мобильные приложения» (8–9 классы)
- «Web-разработка: создай свой идеальный сайт. Обучаемся тонкостям HTML, HTML5, CSS3, SQL, PHP, JavaScript» (8–10 классы)
- «Основы искусственного интеллекта и нейросетей» (10–11 классы, студенты) и другие.