



Посвящается моей
дочери Клэр.
Пусть твое ощущение чуда
не исчезнет никогда



Оглавление

	Как пользоваться этой книгой	9
Глава 1	Добро пожаловать в программирование!	11
	Что такое программа?	
	Основные понятия в программировании	
Глава 2	Основы Scratch	21
	Как установить Scratch	
	Как сохранить проект	
	Осваиваем инструменты	
	Баги	
Глава 3	Создаем движение	35
	Выбор спрайта	
	Оживляем спрайт	
	Собираем все воедино	
	Упражнение 1. Мой первый скрипт	
	Упражнение 2. Вперед, лама, вперед!	
Глава 4	Пишем нескучный код	51
	Меняем костюмы	
	Выбираем фон	
	Звуковые эффекты	
	Создаем движение	
	Упражнение. Хоум-ран — мощный удар по мячу	

Глава 5 Циклы 65

Циклы в Scratch

Упражнение 1. Лепешки тако на перекус

Упражнение 2. Танцевальная вечеринка

Упражнение 3. Берегись падающих звезд!

Циклы в нашей жизни

Исследуем дальше!

Глава 6 Переменные 85

Переменные в Scratch

Упражнение 1. Поговори со спрайтом!

Упражнение 2. Отгадай загадку!

Упражнение 3. Не попадись акуле в пасть!

Переменные в нашей жизни

Исследуем дальше!

Глава 7 Условия 103

Условия в Scratch

Упражнение 1. Назови пароль!

Упражнение 2. Путешествие в космосе

Упражнение 3. Миссия: пончики!

Условия в нашей жизни

Исследуем дальше!

Глава 8 Типы и структуры данных 119

Типы данных и структуры в Scratch

Упражнение 1. Веселый фудтрак

Упражнение 2. Спроси лягушку-волшебницу

Упражнение 3. Игра на угадывание

Типы и структуры данных в нашей жизни

Исследуем дальше!

Глава 9 **Обработка событий** 135

Обработка событий в Scratch

Упражнение 1. Удивительная морская звезда

Упражнение 2. Тук-тук!

Упражнение 3. Крылатый похититель мяча

Обработка событий в нашей жизни

Исследуем дальше!

Глава 10 **Игра продолжается: объединяем все инструменты** 155

Игра продолжается!

Квест с волшебной палочкой!

Переходим к творчеству: создаем свою игру

Напоследок о коде 173

Охота за багами: советы

по устранению неполадок 174

Глоссарий 176

Ресурсы 178

Алфавитный указатель 179

Об авторе 190

Как пользоваться этой книгой

Хочешь узнать, что такое программирование? Эта книга научит тебя думать и решать задачи как программист! Мы познакомимся с основами кодирования и затем освоим первые шаги в использовании Scratch – увлекательного языка программирования. Ты изучишь важные понятия, которые войдут в твой набор инструментов для создания программ. В каждой главе тебе предстоит выполнить по три разных упражнения, каждое из которых будет немного сложнее предыдущего. Когда ты освоишь набор инструментов, мы вместе напишем интересную игру, в которой ты будешь сражаться с огнедышащим драконом, чтобы вернуть волшебную палочку! После этого ты почувствуешь в себе уверенность и будешь готов создавать все, что только сможешь вообразить!

В конце этой книги ты найдешь несколько важных разделов. Если возникнут вопросы, ты всегда можешь заглянуть в раздел «Охота за багами: советы по устранению неполадок» (стр. 174). В нем дается список распространенных ошибок при написании кода и советы, как их устранить. В «Глоссарии» (стр. 176) ты найдешь определения новых для тебя слов из этой книги. Когда ты выполнишь все упражнения и напишешь окончательный вариант игры, загляни в раздел «Напоследок о коде» (стр. 173), чтобы найти классные идеи на будущее!

Не забудь спросить разрешения у родителей или других взрослых, чтобы загрузить Scratch из интернета или открыть его. Помни этот полезный совет по технике безопасности каждый раз, когда заходишь на сайт онлайн.

Хорошая новость: дети великолепно пишут код. Ты умный и любознательный, тебе нравится пользоваться компьютером, верно? Тогда у тебя есть все, чтобы начать.

Итак, ты готов узнать, как писать код, и выучить новый интересный язык? Отлично! Тогда начнем осваивать новые навыки!





1



Добро пожаловать в программирование!

Программировать действительно очень интересно, но для начала давай разберемся, что же это такое?

Программирование – это, по сути, написание инструкций для компьютера. В них мы указываем ему, что нужно сделать. При этом и во время обычной работы мы постоянно даем ему какие-либо команды. Разве ты не говоришь своему компьютеру, что он должен открыть сохраненный доклад и распечатать его для занятий в школе? Да, и кроме этого, чтобы вывести на печать документы, компьютеру приходится выполнять еще множество других команд.

Во-первых, когда ты кликаешь на иконку доклада, компьютер понимает, что нужно открыть именно этот файл и показать его на экране.

Во-вторых, если ты решишь написать несколько предложений, то компьютер должен знать, какую букву отображать при нажатии каждой из клавиш.

И даже простое перемещение курсора мышки по экрану до кнопки печати потребует от компьютера выполнения множества команд.

Эти невидимые инструкции, которым следует компьютер, называются **кодом**. Вот почему про человека, который программирует, говорят, что он «кодит». Он пишет все инструкции, которым следует компьютер. Любое электронное устройство, когда ты им пользуешься, постоянно проверяет свой код и следует всем указаниям, которые кто-то для него запрограммировал. У него нет такого мозга, как у тебя или у меня, поэтому он зависит от кода, который указывает ему действия.

Так почему тебя должен интересовать весь этот код? Ты не видишь, как там все происходит. Можно просто расслабиться и наслаждаться интересной работой, которую выполняет компьютер благодаря коду.

Появлялось ли у тебя когда-нибудь желание, чтобы компьютер сделал что-то еще или то же самое, но *лучше*? Например, чтобы можно было перемещать персонажа в игре про ниндзя не с помощью стрелок, а мышью?

Если ты умеешь **программировать**, то сможешь сам решать, каким командам должен следовать твой компьютер! Ты можешь заставить его вести себя так, как пожелаешь. Ты можешь создавать *сам*. Это что-то вроде разницы между чтением рассказа и его *написанием*. Конечно, читать рассказы приятно, но насколько интереснее самому придумывать идеи и писать *собственную* историю, а потом делиться ею с друзьями?

Если ты научишься писать код, это вовсе не означает, что ты должен стать профессиональным программистом, когда вырастешь. Хотя это вполне можно сделать, если ты захочешь! Программирование полезно знать *каждому*, это помогает научиться разбирать проблемы по частям и решать их.

Создание собственных программ поможет тебе воплощать собственные проекты – начиная с идеи в голове и заканчивая готовой игрой! Программирование поможет тебе лучше понимать, что происходит внутри устройств, которыми ты постоянно пользуешься.

Эта книга – руководство для детей, которые никогда раньше не писали код, но готовы научиться думать как программист.

Что такое программа?

Программа – это список инструкций (или кода), которые сообщают электронному устройству, что делать. Эти инструкции записаны в определенном порядке, чтобы устройство могло понять, какую именно задачу нужно выполнить. Все команды должны быть разбиты на маленькие, четкие шаги.

Представь, что ты пытаешься обучить инопланетянина открывать дверь, которую он никогда раньше не видел. Он не поймет, если просто сказать: «Поверни ручку и толкни ее, чтобы открыть». У инопланетянина возникнет множество вопросов: «Где находится ручка? Чем ее повернуть? Куда мне нужно толкнуть?»

Инопланетянин никогда раньше не был на Земле, поэтому он просто ничего не поймет. Тебе следует разъяснить бедному пришельцу все до мельчайших подробностей, как именно открыть эту дверь. Указания для нашего озадаченного друга могут быть такими.

1. Положи руку на круглый металлический объект, расположенный в середине левой части большого деревянного прямоугольника.
2. Поверни своей рукой металлический объект на 90 градусов вправо.
3. Удерживай металлический объект в том же положении и одновременно толкай другой рукой большой деревянный прямоугольник вперед.
4. Шагни вперед через отверстие.
5. Отпусти круглый металлический объект.

Именно так ведут себя компьютеры. Они напоминают инопланетянина, который не может сделать даже простые вещи без подробных инструкций. Программы – это указания, которые помогают им понять, чего мы, люди, хотим от них. Это означает, что когда ты пишешь программу, то должен исходить из

того, что компьютер абсолютно ничего не знает! Для этого тебе нужно разбить каждое действие на очень маленькие шаги.

Каждая программа состоит из пяти основных частей

ВВОД

Ввод – это информация, которая вводится в компьютер. Представь, что ты решаешь математическую задачу на калькуляторе. Ты напечатал «2+2». Это и есть ввод.

ВЫВОД

Вывод – это информация, которую показывает нам компьютер. Вернемся к нашему калькулятору. Когда ты печатаешь задачу «2+2», то калькулятор показывает «4» – это вывод.

ВЫЧИСЛЕНИЕ

С этим ты точно знаком! **Вычисление** – это операция, которую выполняет компьютер. Для этого, например, калькулятор должен знать, как сложить 2 и 2, чтобы вывести нам правильный результат – «4».

УСЛОВНОЕ ВЫПОЛНЕНИЕ

Условное выполнение означает, что существуют различные пути или условия, которые могут меняться в зависимости от вводимых данных. Если ты введешь «2 + 2», то калькулятор выведет «4». Если ты введешь «5 + 5», то калькулятор выведет «10». Это два отдельных возможных пути решения. Другими словами, условное выполнение – это правила, которым должен следовать компьютер.

ПОВТОРЕНИЕ

При **повторении** многократно воспроизводится набор шагов. Следуя программе, компьютер повторяет набор команд определенное количество раз или до того момента, когда получит определенный результат. Например, калькулятор может повторять команду и продолжать добавлять любые

числа, которые ты нажимаешь на клавишах до момента ввода, допустим, знака «=». Тогда он начинает следовать новым командам, которые нужны для выполнения математической операции в твоей задаче. Калькулятор сотрет числа с экрана и выведет результат – решение математической задачи.

Основные понятия в программировании

Существует много различных языков программирования, но все они используют одни и те же базовые инструменты для решения задач и создания программ. Понятия, которые ты изучишь в этой книге, войдут в твой набор инструментов кодирования. Ты узнаешь, что каждый инструмент может совершать много разных действий, однако некоторые из них подходят больше для каких-либо определенных задач. Например, молотком очень удобно забивать гвозди, но плохо распиливать доски – для этого надо будет взять пилу! Когда ты разберешься с понятиями кодирования, то поймешь, как мыслят программисты, как применять одновременно много разных инструментов для создания более объемных и сложных программ. И тогда ты сам сможешь писать программы!

Основные понятия

ПЕРЕМЕННЫЕ

Переменные – это инструмент, который используется для хранения информации. Ты можешь менять значение этой информации, когда это необходимо. Переменные пригодятся, например, для отслеживания счета в игре.

УСЛОВИЯ

Условия помогают нам создавать условные выполнения, о которых мы говорили выше. Ты можешь придумывать самые разные действия, как в приключенческой истории на выбор! Например, если пользователь кликнет на лягушку, она начнет

прыгать вверх-вниз. Но если он кликнет на пруд, то лягушка нырнет в воду. Это два возможных действия, которые доступны пользователю.

ТИПЫ ДАННЫХ И СТРУКТУРЫ ДАННЫХ

Данные – это просто модное слово для обозначения информации. Типы данных и структуры данных – это способы организации нашей информации в программе, подобные тем, что ты используешь для расстановки предметов на парте.

ОБРАБОТКА СОБЫТИЙ

События – это триггеры для кода, они его запускают. Как только ты вызываешь событие (щелкаешь мышкой на иконку), компьютер начинает следовать набору команд (или коду), которые связаны с этим событием.

ЦИКЛЫ

Циклы нужны для создания повторения, о котором мы говорили выше. Чтобы программа повторяла одни и те же шаги несколько раз, используй цикл. Представь, что ты написал код, который создает прыгающий на экране мяч. Если ты хочешь, чтобы он подпрыгнул сто раз, то будет легче написать цикл «повторить 100 раз», чем дублировать этот код сто раз!

Мы рассмотрели основные понятия кодирования, которые как инструменты использует каждый программист независимо от языка программирования. Если ты немного запутался, то не переживай! Как только начнешь писать код программы, многие из этих терминов станут более понятными.

Говори как программист: слова, которые нужно знать

Мы уже познакомились с инструментами, а теперь давай выучим несколько слов, которые нам пригодятся.

ЗАПУСК

Код – это всего лишь инструкция для компьютера. Компьютер начинает выполнять эти инструкции, только когда мы

запускаем код. Программист может сказать: «Давай **запустим** эту программу и посмотрим, работает ли она».

АЛГОРИТМ

Алгоритм – это любая логическая группа шагов, которая используется для выполнения задачи или решения проблемы. Это очень похоже на определение программы. Но в действительности алгоритм – это *идея*. А программа – фактический код для воплощения этой идеи в жизнь. Следование алгоритму напоминает приготовление по рецепту. Пирог, который ты приготовил по рецепту, – это и есть программа. Программист может сказать: «Мне нужно придумать хороший алгоритм для моей программы».

СКРИПТ

Когда программисты говорят о **скрипте**, они имеют в виду код, написанный на специальном языке программирования. Программист может сказать: «Я работаю над скриптом, который заставит космический корабль двигаться быстрее». Поскольку мы будем использовать специальный язык программирования Scratch, то код, который ты напишешь, можно называть скриптом.

ФУНКЦИЯ

Функция – это кусочек кода, который не запускается, пока ты не сделаешь специальный ввод. Например, калькулятор не запускает код для решения математической задачи, пока ты не нажмешь кнопку «=». До этого момента код представляет собой функцию, которая ожидает своего триггера или события. Программист может сказать: «Эта функция не запустится, пока ты не нажмешь зеленую кнопку».

ОПЕРАТОРЫ

Операторы относятся к специальному типу функции, который позволяет совершать математические вычисления, как, например, сложение и вычитание. Они также могут делать логические сравнения, такие как больше (>) или меньше (<). Программист может сказать: «Если счет становится больше 10, то я хочу, чтобы на экране появилось *“Вы выиграли”*. Я буду использовать оператора для этого».

Почему Scratch?

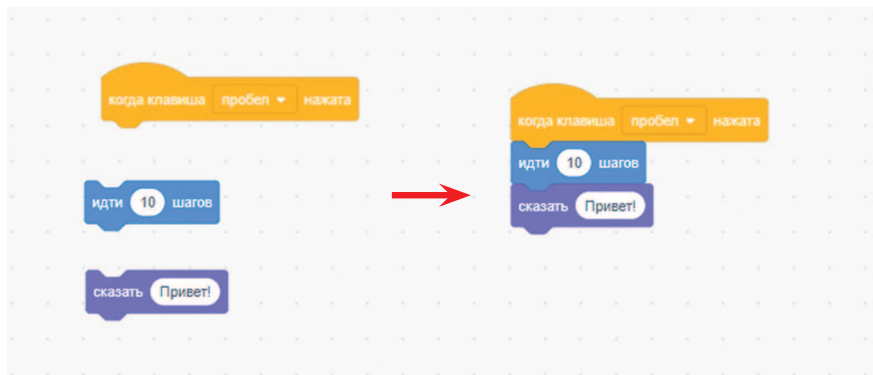
Возможно, ты подумал: «Почему я должен начинать со Scratch?»

Эта книга научит тебя мыслить как программист, поэтому ты сможешь чувствовать себя уверенно, осваивая *любой* другой язык программирования. Для знакомства с программированием мы будем использовать Scratch. Этот язык создали учителя, студенты и исследователи Массачусетского технологического института (MIT).

Мы выбрали Scratch, потому что это веселый язык, который был создан специально для детей. Для написания кода в Scratch используются блоки, которые соединяются как в конструкторе, что делает его *визуальным* языком. Это значит, что ты можешь видеть, как код изменяет программу по мере ее написания, и не нужно долго ждать!

Scratch содержит разнообразное меню блоков для кодирования со всевозможными командами. Каждый кусочек кода – это блок, в котором записана одна команда. Блоки соединяются между собой, образуя длинные столбики. Когда ты соединяешь блоки кода вместе, то создаешь список пошаговых команд, или, другими словами, *программу*.

Люди во всем мире любят Scratch, потому что с ним весело работать. В нем есть много классных персонажей, фонов и музыки для игр.



В командах на рисунках говорится: при нажатии клавиши пробела перемести персонажа на 10 шагов вперед и скажи: «Привет!»

Во многих языках программирования команды печатаются на клавиатуре. Но писать программу в Scratch – это в большей