

РЕШМА САУДЖАНИ

Основательница движения Girls Who Code против гендерного неравенства в технических профессиях

ПРОГРАММИРУЙ!

*самоучитель
для девочек*

БЕСТСЕЛЛЕР THE NEW YORK TIMES И AMAZON

«РЕШМА САУДЖАНИ И ДВИЖЕНИЕ GIRLS WHO CODE МЕНЯЮТ ЛИЦО СОВРЕМЕННЫХ ТЕХНОЛОГИЙ С КАЖДОЙ НОВОЙ МАЛЕНЬКОЙ ЧИТАТЕЛЬНИЦЕЙ. ЭТА КНИГА – ВОЗМОЖНОСТЬ ДЛЯ ВСЕХ ДЕВЧОНОК ПРИСОЕДИНИТЬСЯ К ДВИЖЕНИЮ ЗА РАВЕНСТВО И ЛУЧШЕЕ БУДУЩЕЕ».

ШЕРИЛ СЭНДБЕРГ,
ИСПОЛНИТЕЛЬНЫЙ ДИРЕКТОР FACEBOOK

РЕШМА САУДЖАНИ

ПРОГРАММИРУЙ!

*самоучитель
для девочек*

БОМБОРА™

Москва 2018

УДК 004.42(075)
ББК 32.973.2-018я7
С21

Girls Who Code: Learn to Code and Change the World
Reshma Saujani

© 2017 by Girls Who Code Inc. This edition published by arrangement
with InkWell Management LLC and Synopsis Literary agency

Сауджани, Решма.
С21 Программируй! Самоучитель для девочек / Решма Сауджани ;
[пер. с англ. Е. Кузьминой]. — Москва : Эксмо, 2018. — 176 с. —
(Программирование для детей).

Своей книгой Решма Сауджани, основательница движения «Girls Who Code», которое борется за гендерное равенство в технических профессиях, помогает маленьким читательницам узнать множество интересных и воодушевляющих фактов о программировании. Основы написания кода и работы с компьютером, истории успешных женщин-программистов, краткое введение в языки программирования и возможность создать свою первую программу, а также сотни других полезных вещей, рассказанных понятным языком в дружелюбной манере, — все это вы найдете в книге «Программируй! Самоучитель для девочек».

УДК 004.42(075)
ББК 32.973.2-018я7

Все права защищены. Книга или любая ее часть не может быть скопирована, воспроизведена в электронной или механической форме, в виде фотокопии, записи в память ЭВМ, репродукции или каким-либо иным способом, а также использована в любой информационной системе без получения разрешения от издателя. Копирование, воспроизведение и иное использование книги или ее части без согласия издателя является незаконным и влечет уголовную, административную и гражданскую ответственность.

Научно-популярное издание
ПРОГРАММИРОВАНИЕ ДЛЯ ДЕТЕЙ

Решма Сауджани
ПРОГРАММИРУЙ!
Самоучитель для девочек

Главный редактор *Р. Фасхутдинов*. Ответственный редактор *Е. Истомина*. Редакторы *Н. Гринчик, Е. Науменко*
Литературный редактор *Н. Калининченко*. Научный редактор *А. Журба*. Младший редактор *Е. Минина*
Художественный редактор *А. Гусев*. Компьютерная верстка *Н. Зенков*. Корректор *Е. Павлович*

ООО «Издательство «Эксмо»

123308, Москва, ул. Зорге, д. 1. Тел.: 8 (495) 411-68-86.

Home page: www.eksmo.ru E-mail: info@eksmo.ru

Өндіруші: «ЭКМО» АҚБ Баспасы, 123308, Мәскеу, Ресей, Зорге көшесі, 1 үй. Тел.: 8 (495) 411-68-86.

Home page: www.eksmo.ru E-mail: info@eksmo.ru Тауар белгісі: «Эксмо»

Интернет-магазин : www.book24.ru

Интернет-дукен : www.book24.kz

Импортер в Республику Казахстан ТОО «РДЦ-Алматы».

Қазақстан Республикасындағы импорттаушы «РДЦ-Алматы» ЖШС.

Дистрибутор и представитель по приему претензий на продукцию, в Республике Казахстан: ТОО «РДЦ-Алматы»

Қазақстан Республикасында дистрибутор және өнім бойынша арыз-талаптарды қабылдаушының өкілі «РДЦ-Алматы» ЖШС,

Алматы қ., Домбровский көш., 3-а, литер Б, офис 1. Тел.: 8 (727) 251-59-90/91/92; E-mail: RDC-Almaty@eksmo.kz

Өнімнің жарамдылық мерзімі шектелмеген.

Сертификация туралы ақпарат сайтта: www.eksmo.ru/certification

Сведения о подтверждении соответствия издания согласно законодательству РФ

о техническом регулировании можно получить на сайте Издательства «Эксмо» www.eksmo.ru/certification

Өндірген мемлекет: Ресей. Сертификация қарастырылмаған

Подписано в печать 21.05.2018. Формат 70x100^{1/16}.

Печать офсетная. Усл. печ. л. 14,26. Тираж экз. Заказ

ISBN 978-5-04-093243-6



9 785040 932436 >

ISBN 978-5-04-093243-6



В электронном виде книги издательства вы можете
купить на www.litres.ru

ЛитРес:
один клик до книги



© Кузьмина Е., перевод на русский язык, 2018
© Оформление. ООО «Издательство «Эксмо», 2018

Посвящается всем юным
программисткам —
нынешним и будущим.

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ 5

HELLO, WORLD! 8

1 ПОЧЕМУ ПРОГРАММИРОВАНИЕ? 1 5

2 КАК ОБЩАТЬСЯ СО СВОИМ КОМПЬЮТЕРОМ 3 3

3 СОБИРАЕМ ВСЕ ВМЕСТЕ 4 5

4 С ЧЕГО НАЧАТЬ? 6 1

5 РАЗГАДАЙ КОД 7 7

6 ОТЛАДКА 9 9

7 КОМПЬЮТЕРНЫЕ ИГРЫ 1 1 1

8 ЦИФРОВОЕ ИСКУССТВО И ДИЗАЙН 1 2 3

9 РОБОТЫ 1 3 5

10 САЙТЫ, МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ
И ИНТЕРНЕТ-БЕЗОПАСНОСТЬ 1 4 5

ЗАКЛЮЧЕНИЕ 1 5 8

СЛОВАРЬ ТЕРМИНОВ 1 6 4

БЛАГОДАРНОСТИ 1 7 0

АЛФАВИТНЫЙ УКАЗАТЕЛЬ 1 7 1

ПРЕДИСЛОВИЕ

Я ОТЧЕТЛИВО ПОМНЮ ДЕНЬ, КОГДА У НАС ДОМА ПОЯВИЛСЯ ПЕРВЫЙ компьютер. Мне было 9 лет. Он шуршал в углу, а я все не знала, как к нему подступиться. Наша дружба началась с компьютерных игр — и вскоре я сама не заметила, как не могла без него ни дня. В 13 лет я научилась программировать, а в 24 — создала свою компанию по обучению детей программированию.

В школе я программировала лучше всех в классе, но на уроках, посвященных этому предмету, мне было скучно: мы создавали слишком простые и бессмысленные программы вроде Hello, world. Я же мечтала защищать мир от хакерских атак и писать сложные алгоритмы, которые никто, кроме меня, не поймет.

Создавая свою школу программирования, я точно знала, что дети должны учиться в реальной среде и на интересных проектах. В этом смысле философия, описанная Решмой Сауджани в этой книге, во многом совпадает с нашей философией. Но Решма идет еще дальше. Она преподает азы программирования и алгоритмики на понятных примерах из жизни, при этом не используя ни строчки кода! Тем самым Решма стремится устранить огромный гендерный разрыв в сфере изучения программирования, произошедший в последние десятилетия.

Надо отметить, что девочки совершенно не нуждаются в специфическом образовательном подходе. Хотя большинство наших студентов мальчики, девочки просто созданы для программирования. Они всегда презентуют серьезные проекты, потому что очень внимательно и усердно во всем разбираются. Кто-то любит рисовать, кто-то — анимировать, но даже код дается девочкам проще, потому что они усидчивы, дисциплинированы и внимательно слушают преподавателя.

Почему же тогда и в крупных, и в малых IT-компаниях женщины представляют собой меньшинство? Почему с каждым годом процент женщин, задействованных в IT-индустрии, продолжает стремительно падать? Если в 1995 году женщинами были 37% компьютерных ученых, то в 2018 году — всего 24%. Если ничего не предпринять, через десять лет эта цифра упадет еще на 2%!

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

В своей книге Решма Сауджани пишет о том, что причиной такой ситуации становятся стереотипы в нашем обществе. Девочка, решившая связать свою жизнь с программированием, почти наверняка столкнется с неприятием своего выбора. Она будет периодически слышать, что работа программиста — неженская, что ей не разобратся во всех тонкостях, что программисты — это исключительно парни в толстовках, что девчонке невозможно стать лучшей в этой среде. И хотя все это глупости и предрассудки, многие девочки, не нашедшие вовремя поддержки и примеров для подражания, отказываются от своего желания стать программистом и выбирают что-то более привычное и социально одобряемое.

Эта книга помогает разрушить комплексы и стереотипы насчет программирования, делает компьютерную науку доступной, понятной и привлекательной. Ведь даже бисероплетение и вышивание, которыми занимаются многие девочки, — это своего рода программы. Если девочки справляются с этим, значит, и код осилят!

Всего на 170 с небольшим страницах своей удивительной книги Решма Сауджани успевает рассказать о программировании в компьютерных играх, цифровом искусстве, дизайне, о создании сайтов и мобильных приложений, робототехнике, информационной безопасности. Построить алгоритм почти так же легко, как приготовить шоколадный маффин или выучить движение для танца. А тема программных интерфейсов рассмотрена в книге на примере приложения, созданного учениками школы, — оно помогает выбрать прическу по погоде.

Важная вдохновляющая и мотивирующая составляющая книги — вставки об известных женщинах-программистах, изменивших мир.

Вы все еще думаете, что у вас нет способностей к кодированию? Что программирование не подходит девчонкам? Что у вас аллергия на математику? Или что мечты стать женщиной-лидером во главе IT-компании несбыточны? Книга «Программируй! Самоучитель для девочек» — то, с чего стоит начать, чтобы расширить свое сознание, набраться храбрости и найти единомышленников.

*Дарья Абрамова, основатель и генеральный директор
Школы цифрового творчества «Кодабра»*

НАШИ ПРОГРАММИСТКИ

Анна



ДЕНЬ РОЖДЕНИЯ: 20 МАЯ
УВЛЕЧЕНИЯ: химия, музыка, компьютерные игры, смайлики, пробовать что-то новое

София



ДЕНЬ РОЖДЕНИЯ: 13 НОЯБРЯ
УВЛЕЧЕНИЯ: спорт, спортивные штаны, сидеть с детьми, дизайн ногтей, селфи

Майя



ДЕНЬ РОЖДЕНИЯ: 3 ИЮНЯ
УВЛЕЧЕНИЯ: писать, рисовать, мода, крупные украшения, давать советы

Саша



ДЕНЬ РОЖДЕНИЯ: 26 ФЕВРАЛЯ
УВЛЕЧЕНИЯ: выпечка, театр, чтение, серфинг, пародировать

Гуля



ДЕНЬ РОЖДЕНИЯ: 22 АВГУСТА
УВЛЕЧЕНИЯ: робототехника, садоводство, хоккей на траве, мастерить что-нибудь, проводить время со старшей сестрой

HELLO, WORLD!

Я РЕШМА, основательница «Программирования для девочек».

Наша организация помогает девочкам средних и старших классов научиться программированию, узнать, как устроены компьютеры и электронные устройства и какие новые поразительные идеи, навыки и возможности открываются для тех, кто умеет программировать.

Таких идей и возможностей тысячи, уж поверь.

Скажу кое-что по секрету: **ЕЩЕ НЕСКОЛЬКО ЛЕТ НАЗАД Я БОЯЛАСЬ УЧИТЬСЯ ПРОГРАММИРОВАТЬ.**

Я занимаюсь правом и политикой. Я была помощницей общественного адвоката Нью-Йорка, а в 2010 году стала первой американкой



HELLO, WORLD!

южноазиатского происхождения, которая баллотировалась в Конгресс¹. Мне всегда нравилось заводить новые знакомства и помогать окружающим, поэтому я занялась политикой. С детства я хотела сделать что-то такое, что изменит жизнь людей к лучшему, но даже не думала, что добьюсь этого с помощью компьютеров и программирования.

Во время предвыборной кампании я часто посещала нью-йоркские школы и заметила одну вещь: компьютерные классы были полны мальчиков, которые учились программировать и изобретать технические новшества — **НО ДЕВОЧЕК ТАМ ПРАКТИЧЕСКИ НЕ БЫЛО!**

Куда же они подевались?

Мне это показалось несправедливым. Я знаю, что большинство наших университетских выпускников и половина работающего населения — женщины. Но когда дело доходит до информатики, изучения компьютеров и самых разных способов их применения, женщины остаются в стороне (по крайней мере, в нью-йоркских школах). И это уже проблема.

К 2020 году в компьютерной индустрии будет 1,4 миллиона рабочих мест с самыми высокими зарплатами и стремительными карьерами. **НО ЖЕНЩИНАМ ДОСТАНЕТСЯ ВСЕГО 4% ОТ ОБЩЕГО КОЛИЧЕСТВА.**

Четыре процента! Этого не хватит даже на сектор на круговой диаграмме.

По-моему, это никуда не годится. Девочки упускают возможность получить работу будущего только потому, что не учатся программированию.

Почему так? Почему в этих классах так мало девочек?

ПРОБЛЕМА

И я задумалась: а почему же я сама никогда не училась программированию?

¹ Конгресс — высший орган законодательной власти США. — *Примеч. пер.*

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Не то чтобы у меня не было возможности изучать математику и компьютеры; как-никак мой отец — инженер. Когда я была еще девочкой, он частенько делился со мной научными идеями и обожал внезапно задать мне математическую задачку, обычно за ужином. Мне было сложно дать ответ: нередко я его знала, но не могла быстро сообразить. И когда я молча сидела, глядя через стол на отца, то чувствовала себя недостаточно умной. Ужин превращался в пытку, и я начала думать, что у меня нет математических способностей.

Потом я стала бояться и избегать математики, как и любого предмета, для которого, как мне казалось, нужны математические знания: программирования, статистики, инженерии. Вместо этого я сосредоточилась на истории и письме, где чувствовала себя комфортнее и знала, что справлюсь.

Тогда я этого не понимала, но я была не одна. Тысячи девочек всех возрастов говорят мне то же самое: у них «нет способностей» к математике или естественным наукам. Они говорят, что боятся предметов, которые кажутся им слишком техническими, вроде программирования. А если и не боятся, то думают, что информатика не для них, а для мальчиков-одиночек, которым нравится целыми днями сидеть за компьютером.

Так вот, я открою тебе еще один секрет: **ЭТО ПОЛНЕЙШАЯ ЧЕПУХА НА ПОСТНОМ МАСЛЕ!**

С раннего детства стереотипы, социальные сигналы, а порой и учителя внушают нам, что точные науки — естественные науки, технология, инженерия и математика — «не для тебя». Эти сигналы встречаются на каждом шагу. Можно зайти в популярный магазин, продающий одежду для девочек-подростков, и купить футболку с надписью «У МЕНЯ АЛЛЕРГИЯ НА АЛГЕБРУ». Или посмотреть один из бесчисленных сериалов, где программист — это парень в толстовке с капюшоном, в одиночестве сидящий за компьютером в своем подвале.

HELLO, WORLD!

И мы, девочки, слушаемся. Эти негативные стереотипы и отсутствие примеров для подражания выражаются в числах. Большинство учениц средней школы считают, что карьеры, связанные с точными науками, не для них. В старших классах девочки называют инженерию, работу с компьютерами и математику в числе самых неинтересных профессий.

Девочек отваживают от точных наук раньше, чем они успевают понять, нравятся ли им эти предметы, и, что гораздо более важно, раньше, чем они успевают понять, что эти предметы им отлично даются.

СЛЕДУЮЩИЙ ШАГ

В 2010 году я проиграла выборы в конгресс. Это было непросто, ведь обычно мне всегда все удавалось. Я долго не могла решить, что делать дальше. Но затем я поняла кое-что, что навсегда изменило мою жизнь: решение баллотироваться в конгресс потребовало от меня огромного мужества. Мне пришлось стать храброй, и пусть у меня ничего не вышло, я знала, что попыталась. Я вышла из своей зоны комфорта и занялась чем-то другим, новым, пугающим. А если бы я поступила так раньше, за ужином или в школе? Вдруг бы я поняла, что обожаю программирование, или математику, или естественные науки?

Поэтому я решила, что должна призвать свое мужество и использовать свой опыт, чтобы дать новые возможности следующему поколению девушек.

Я решила учить девочек программированию. Мой эксперимент начался с класса из 20 девочек в Нью-Йорке. Я убедила друга, чтобы он разрешил мне воспользоваться конференц-залом в его компании, и отправилась по домам набирать первых учениц. Я не знала, как отреагируют люди, но понимала, что должна попытаться.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

СЕЙЧАС «ПРОГРАММИРОВАНИЕ ДЛЯ ДЕВОЧЕК» ВЫРОСЛО В НАСТОЯЩЕЕ ДВИЖЕНИЕ. Мы организовали 100 летних курсов и тысячи секций для учениц средних и старших классов. С нами десятки тысяч девочек из каждого американского штата.



И ЗНАЕШЬ ЧТО?

Девочки — отличные программисты!

Они могут разрабатывать потрясающие вещи!

И делают это с удовольствием!

Из этой книги ты узнаешь о моих любимых разработках, сделанных реальными девочками: от игры, которая помогает девочкам понять, что они красивы, до системы освещения, создающей подсветку в такт музыке. Когда ты учишься программировать, ты обретаешь суперсилу, средство, чтобы изменить что-то в своем окружении. С помощью *своего*

HELLO, WORLD!

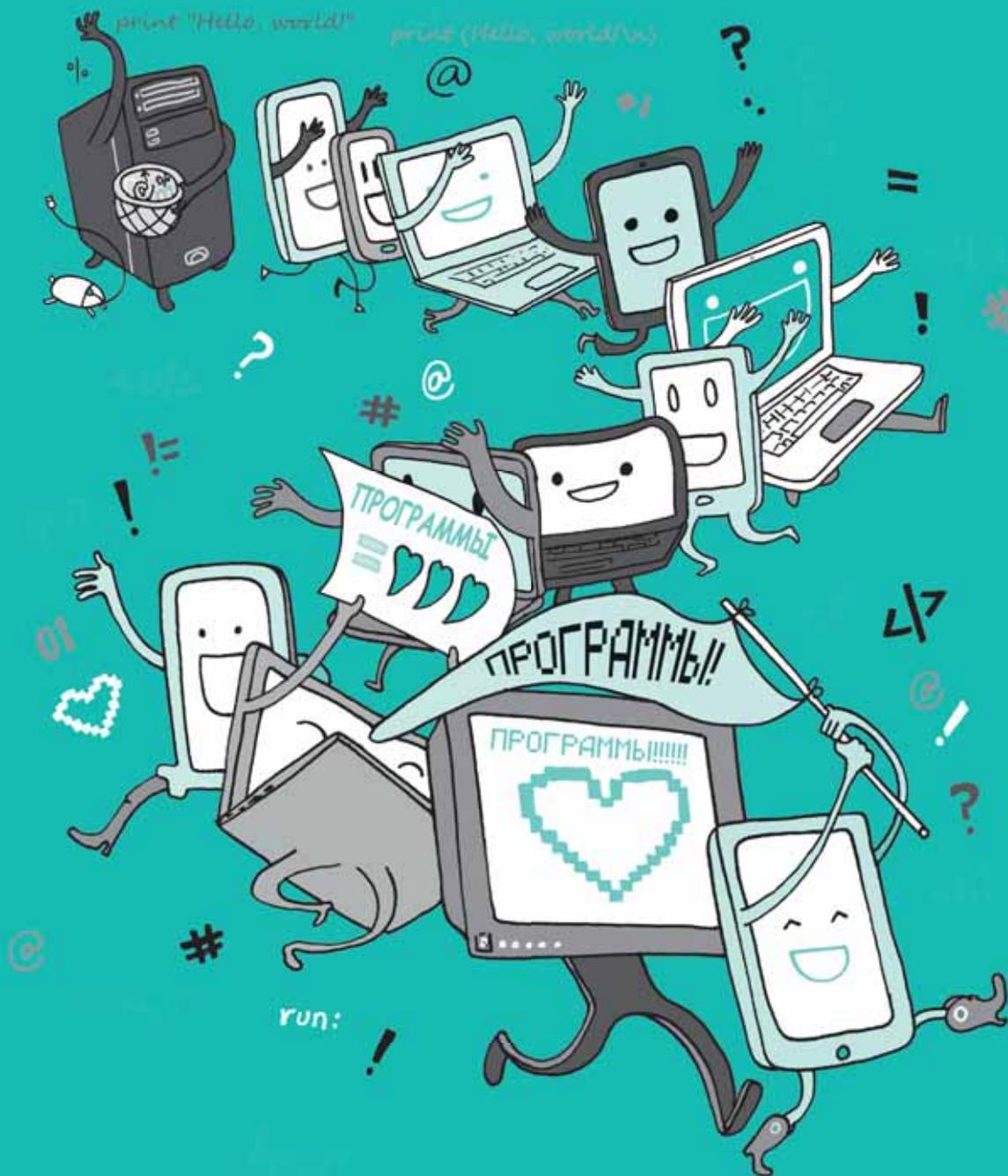
собственного голоса, *своего* разума и *своих* навыков ты решаешь проблемы, чтобы помочь своей стране и создать новый, лучший мир для всех и каждого из нас.

И конечно, заводишь новых друзей и веселишься от души.

ТАК ЧЕГО ЖЕ ТЫ ЖДЕШЬ?

Эта книга поможет тебе стать девочкой, которая умеет программировать и создавать удивительные вещи. Ты изучишь те же основы, которые мы преподаем на наших курсах «Программирование для девочек», познакомишься с новыми проектами и научишься программировать легко и весело на примерах игр, рисования и дизайна, роботов, сайтов и мобильных приложений, а также сетевой безопасности. А еще мы расскажем о женщинах и девочках, использующих свои навыки программирования, чтобы создавать удивительные, вдохновляющие вещи. Уверена, ты немедленно захочешь создать что-то свое и присоединиться к нашему девичьему движению, охватившему весь мир.

ГОТОВА? ТОГДА СОБЕРИСЬ С ДУХОМ И ВПЕРЕД!





ПОЧЕМУ ПРОГРАММИРОВАНИЕ?

ДОБРО ПОЖАЛОВАТЬ!

Да-да, ты, читающая сейчас эту книгу! Здорово, что ты решилась.

Почему ты решила заняться программированием?

Может быть, ты интересуешься **ИНФОРМАТИКОЙ**¹ и хочешь узнать, что еще она может тебе дать?

Если так, то это отлично!

¹ Если слово выглядит **ВОТ ТАК**, то ты можешь посмотреть, что оно значит, в словаре в конце этой книги!

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Может, кто-то из твоих родителей, дедушка, бабушка или учитель сказал, что программирование пригодится тебе в будущем, и записал тебя на курсы, но ты не уверена, что это для тебя.

Это тоже неплохо.

Может, ты ничего не знаешь о программировании — тебе просто понравилась обложка.

И это уже что-то!

Как бы то ни было, мы рады, что ты выбрала эту книгу, и рады, что ты с нами. Здесь собралась отличная компания!

Первое, что нужно знать: программирование — это не только компьютеры.

Программирование — это веселье.



ПОЧЕМУ ПРОГРАММИРОВАНИЕ?

И совместная работа с друзьями.

Программировать могут *все* (а не только мальчики).

Значит, ты сможешь придумывать, создавать и изобретать новые занятные приспособления для всего, чем интересуешься.

ПРАВДА?
НАПРИМЕР?



Например, превратишь игрушечную машинку в робота. Или создашь **САЙТ**, чтобы предлагать свои услуги по выгулу собак. А может, разработаешь **ДИЗАЙН** умного браслета для напоминания, что пора делать уроки или садиться за пианино. Или напишешь приложение, которое будет отмечать твое время на пробежках. А как насчет интерактивного светомузыкального дисплея для следующей постановки в школе? Или светодиодного ободка, меняющего цвет, чтобы лучше сочетаться с твоим нарядом?



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Все это, и не только, ты сможешь сделать, если научишься программировать.

И скажу тебе кое-что еще: самый большой и страшный секрет **КОМПЬЮТЕРНОГО ПРОГРАММИРОВАНИЯ** в том, что по сути программирование — это решение задач! Написание собственно программ — очень малая его часть. В основном ты будешь рассуждать и планировать, и ты уже умеешь делать это.

Итак, начнем.

Во-первых, кто мне может сказать, что такое программирование?



Я ЗНАЮ! ЭТО КОГДА ТЫ ГОВОРИШЬ КОМПЬЮТЕРУ, ЧТО НАДО ДЕЛАТЬ.

Верно.

Проще говоря, программирование — это написание команд, указывающих компьютеру, что нужно делать, на том языке, который понятен компьютеру.

Чтобы давать компьютеру указания, существуют сотни разных языков программирования. Какой из них выбрать, зависит от того, какую задачу ты решаешь. Когда ты учишься программировать, ты учишься «говорить» на одном из этих языков, чтобы общаться с компьютером напрямую.

НУ ДА, НО РАЗВЕ МЫ НЕ ГОВОРИМ НАШИМ КОМПЬЮТЕРАМ, ЧТО НАУЖНО ДЕЛАТЬ, КОГДА ЩЕЛКАЕМ КНОПКОЙ МЫШИ И БРОДИМ ПО РАЗНЫМ МЕНЮ И ПРИЛОЖЕНИЯМ? ЗАЧЕМ МНЕ ЕЩЕ И ПРОГРАММИРОВАНИЕ?



ПОЧЕМУ ПРОГРАММИРОВАНИЕ?

Потому что программирование — это удивительный инструмент. С его помощью ты сможешь делать такое, о чем даже не мечтала. Конечно, ты уже можешь кое-что делать на своем компьютере, планшете или смартфоне, но только потому, что кто-то уже придумал, как создать такое приложение или программу, и написал **КОД** приложения. Код, написанный человеком, создал иконки, кнопки и ярлыки, чтобы ты могла управлять своим устройством. Программное обеспечение позволяет компьютеру работать. Но пользоваться программами — не то же самое, что программировать.

Еще одна причина учиться программированию: оно поможет тебе понять технологию будущего, подскажет, как проектировать ее и работать с ней.

ПРОГРАММНОЕ И АППАРАТНОЕ ОБЕСПЕЧЕНИЕ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ — это набор программ и приложений. Они обеспечивают работу компьютера. Программное обеспечение состоит из кода, его разрабатывают и пишут программисты. К программному обеспечению относятся все программы: приложения, в которых ты накладываешь на фото смешные фильтры; игры, где ты заботишься о виртуальном щенке или сражаешься со злодеями, а также компьютерные программы, при помощи которых ты пишешь сочинения.

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ — это механические части самого компьютера: экран, клавиатура, камера и т. п.; все это проектируют и создают инженеры. Твой смартфон или планшет также можно назвать аппаратным обеспечением.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Компьютеры есть почти везде — в машинах, играх, медицинских приборах, одежде... Бывают даже умные зубные щетки!

Если сейчас предмет никак не связан с компьютером, то, вполне возможно, будет связан через несколько лет. Каждое электронное устройство зависит от человека, который пишет программу и объясняет машине, что ей делать и как.



ТО ЕСТЬ ПИСАТЬ ПРОГРАММЫ — ЭТО ОЧЕНЬ ВАЖНАЯ РАБОТА?

Да! Без программистов компьютер был бы просто большой коробкой. Сегодняшние компьютеры способны на многое, но все равно они машины, которые в определенный момент нуждаются в указаниях человека.

ЧТО ДЕЛАЮТ КОМПЬЮТЕРЫ

Вспомни микроволновку у себя на кухне: она ведь не начнет ни с того ни с сего разогревать остатки макарон с сыром, потому что тебе так захотелось. Надо самой поставить тарелку в микроволновку, закрыть дверь, настроить таймер и нажать кнопку «Старт». Машина выполнит работу, которую ты ей задала, и ты получишь свои горячие макароны. Все это

ПОЧЕМУ ПРОГРАММИРОВАНИЕ?

происходит потому, что ты решила, что голодна, и запрограммировала машину сделать то, что было нужно тебе.

В простейшем варианте точно так же работает и компьютер. Ты даешь ему что-то, говоришь, что нужно выполнить, и получаешь результат.

Последовательность выглядит так:



ВВОД ⇒⇒ ОБРАБОТКА ⇒⇒ ВЫВОД

Существует множество способов ввести информацию в компьютер. Самый очевидный — при помощи клавиатуры (именно так я печатала эту книгу). На клавиатуре можно набирать цифры и буквы. В примере с книгой **ВЫВОД** — это напечатанный текст, то есть книга, которую ты читаешь. Но клавиатура не единственное устройство **ВВОДА**. Можно воспользоваться цифровой ручкой, видеокамерой, микрофоном, сканером или сенсором, чтобы ввести в компьютер информацию, с которой он может сделать почти все, что угодно.

Информация — это не всегда факты или числа. Информацией называются любые данные: музыка, видео, рисунок или фотография. С помощью программного обеспечения можно монтировать фильмы, управлять звуковыми дорожками, регулировать цвет, добавлять яркость и тени, анимировать игры, накладывать фильтры на селфи и загружать их в свой профиль. Все это возможно благодаря людям, которые придумали программы.

Программа — это то, что компьютер сделает с информацией, которую ты введешь. Фактически это работа, которую компьютер выполняет по твоей команде, или **ПРОЦЕСС**, который применяется к введенным данным.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Результатом будет **ВЫВОД**. Когда компьютер выполнит программу, ты получишь усовершенствованное фото, документ Word, результат вычисления, смонтированный фильм или анимацию.

Каждое умное устройство или компьютерная программа являются результатом работы многих людей. Сначала у одного программиста возникла идея, он скооперировался с другими программистами, и вместе они решили, что именно будет делать компьютер. Они подумали о том, чтобы человеку, работающему за компьютером, было понятно и удобно. Программисты разработали дизайн программы, написали код, протестировали его, чтобы убедиться, что он работает, — и *вуаля!*

Благодаря работе программистов ты можешь украшать свою ленту в социальных сетях бесконечными вариациями замечательных смайликов.



Научившись программировать, ты сама сможешь делать то же самое. Возможно, у тебя уже есть идеи для полезных и важных программ, которые могут изменить мир.

Что скажешь, смайлик? 😊



ПОЧЕМУ ПРОГРАММИРОВАНИЕ?

Да! Это и есть программирование: сила, которая заставляет компьютеры работать на нас.

Технологии постоянно развиваются, и суперсила программиста очень пригодится тебе, чтобы работать с устройствами будущего. Возможно, когда ты поступишь в университет, технологии будут сильно отличаться от тех, какими ты пользуешься сейчас.



НЕУЖЕЛИ ВСЕ ТАК
БЫСТРО МЕНЯЕТСЯ?

Да. Уж поверь, в средних классах я болтала с подружками по стационарному телефону, висящему на кухонной стене, и собирала плей-листы на кассетах.



МОЖЕШЬ ПОГУГЛИТЬ, ЧТО ЭТО ТАКОЕ!

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

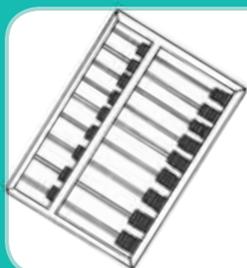
КОМПЬЮТЕРЫ ТОГДА И СЕЙЧАС

Чтобы увидеть, насколько быстро развиваются технологии, давай посмотрим, как появились компьютеры. Ты наверняка удивишься, когда узнаешь, что многие мыслители и изобретатели в области информатики — женщины, о которых почти не пишут в книгах по истории!



Первые компьютеры использовались, как легко догадаться по названию (computer от англ. compute — «вычислять»), для расчетов. Приборы для вычисления и подсчета известны людям с давних времен. Древние цивилизации использовали их, чтобы запоминать большие числа, прокладывать курс для кораблей и изучать ночное небо. Но понадобилось несколько столетий технических новшеств, чтобы появилось что-то похожее на современный компьютер.

ИСТОРИЯ КОМПЬЮТЕРОВ, ЧАСТЬ 1. ПЕРВЫЕ КОМПЬЮТЕРЫ



Абак — счетная доска с камешками. Абак был изобретен в Древнем Вавилоне и распространился по Древнему миру, в том числе применялся в Китае.

**3 ТЫСЯЧИ ЛЕТ
ДО Н. Э.**

Кипу — счетная система инков. Инки записывали числа, завязывая узелки на веревке.

1400 г.

**НУЛЕВОЙ
ГОД**

**35–20 тысяч лет
ДО Н. Э.**

Кости Лебомбо и Ишанго. Древнейшими известными счетными приборами являются найденные в Африке кости бабуина с насечками.

79 г. н. э.

Антикитерский механизм. В Древней Греции с помощью этого устройства рассчитывали месяцы и движение небесных тел.

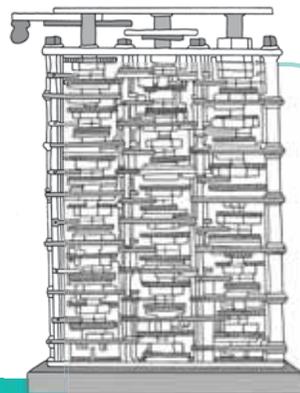
1622 г.

Логарифмическая линейка. После того как шотландский математик Джон Непер изобрел логарифм и очень ускорил математические вычисления, математик Уильям Отред создал логарифмическую линейку. Она была самым популярным вычислительным устройством до появления электронного калькулятора!

ПОЧЕМУ ПРОГРАММИРОВАНИЕ?

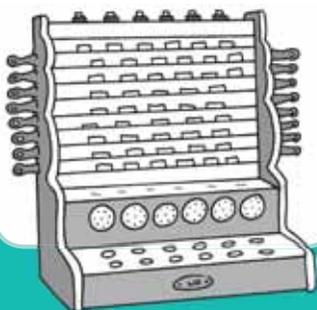
В 1822 году британский математик и инженер по имени Чарльз Бэббидж создал первый полностью механический компьютер. Машина принимала введенные числа, выполняла вычисления и выводила результат. Она состояла из металлических шестеренок и рычагов и называлась «разностная машина». В то время сложные вычисления в области грузоперевозок, промышленности и банковского дела велись в основном по печатным таблицам. Такие расчеты были слишком медленными, к тому же в таблицах было много ошибок, что не совсем приятно, когда рассчитываешь вес грузового контейнера или крупную финансовую операцию. Быстрый и точный механический калькулятор Бэббиджа должен был решить эту проблему.

Хотя машина так и не была закончена, проект произвел фурор. Позднее Бэббидж на его основе создал схему более совершенной аналитической машины. Она стала основой современного компьютера и показала, что нужен инструмент, который может выполнять вычисления быстро, точно и без свойственных человеку ошибок.



Счетная машина. Первое механическое устройство для счета создал немецкий мыслитель Вильгельм Шиккард. Машина Шиккарда могла складывать и вычитать шестизначные числа и использовалась для расчета астрономических таблиц.

1623 г.



Перфокарты. Систему перфокарт придумал Жозеф-Мари Жаккар. С помощью перфокарт можно было автоматизировать самые разные машины, такие как музыкальные шкатулки, механические пианино, счетные машины и ткацкие станки.

1801 г.

Разностная машина, которую спроектировал и частично построил Чарльз Бэббидж, считается предшественницей современного компьютера.

1822 г.

1674 г.

Арифмометр Лейбница. Счетная машина, которую изобрел немецкий философ Готфрид Вильгельм Лейбниц, могла не только складывать и вычитать, но и делить и умножать.

1820 г.

Арифмометр. Первый калькулятор массового производства изобрел Шарль Ксавье Тома де Кольмар.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК



Первый в мире программист **АДА ЛАВЛЕЙС**

Августа (Ада) Байрон Лавлейс — дочь лорда Байрона, знаменитого британского поэта-романтика. Она вошла в историю как первый программист в мире. Ада была очень умна и, в отличие от большинства женщин своего времени, интересовалась

математикой. Уже в 12 лет Ада создала подробный чертеж парового летательного аппарата. В 17 она познакомилась с Чарльзом Бэббиджем, который стал ее другом и наставником на всю жизнь. В 1843 году, когда Аде было 27, Бэббидж попросил ее написать комментарии к его проекту аналитической машины. В один из комментариев, где говорилось, как работать с машиной, Ада добавила пошаговую инструкцию. Тогда никто этого не понял, но эта инструкция, описывающая, как «запрограммировать» машину, чтобы она выполняла вычисление, была первой в мире компьютерной программой. В честь Ады Лавлейс назван язык программирования Ada. С его помощью управляют космическими спутниками.

ЭЛЕКТРОННЫЕ КОМПЬЮТЕРЫ

Прошло еще 124 года, полных вдохновения, изобретательства, проб, ошибок и технических прорывов, прежде чем был создан электронный компьютер, способный делать то, о чем Бэббидж мечтал много лет назад. Электронный числовой интегратор и вычислитель, или ЭНИАК, был первым полностью рабочим электронным компьютером общего назначения без движущихся механических частей. (Уф, насилу выговоришь!) Джон Преспер Эккерт и Джон Мокли из Пенсильванского университета создали его для американских военных во время Второй мировой войны.

ПОЧЕМУ ПРОГРАММИРОВАНИЕ?

ИНТЕРЕСНЫЙ ФАКТ: ученые считают, что ЭНИАК за первые десять лет выполнил больше вычислений, чем человечество за всю свою историю до появления компьютера. В общем, это очень много расчетов.

НАСТОЯЩИЕ КОМПЬЮТЕРЫ — ЖЕНЩИНЫ

Изобретение ЭНИАК произвело революцию в информатике, но мало кому известно, что первыми программистами электронного вычислителя были шесть девушек. Чтобы компьютер мог выполнять вычисления, эти высококвалифицированные работницы вручную вводили **ДАННЫЕ** и программировали операции. Для этого им приходилось загружать перфокарты, устанавливать переключатели и соединять кабели. Как и множество замечательных женщин в истории, тогда они не получили признания за свою новаторскую работу.

Поэтому давай вспомним их сейчас! Спасибо вам:

Фрэнсис Билас Спенс (1922–2013),

Джин Дженнингс Бартик (1924–2011),

Мэрлин Уэскофф Мельцер (1922–2008),

Кэтлин (Кей) Макналти Мокли Антонелли (1921–2006),

Фрэнсис Элизабет (Бетти) Холбертон (1917–2001),

Рут Лихтерман Тейтельбаум (1924–1986).

УХ ТЫ! ЕСЛИ СТОЛЬКО ЖЕНЩИН ПОНАДОБИЛОСЬ, ЧТОБЫ ЗАПРОГРАММИРОВАТЬ ОДИН КОМПЬЮТЕР, НАВЕРНОЕ, НИКТО, КРОМЕ НИХ, НЕ МОГ С НИМ УПРАВИТЬСЯ.



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Верно подмечено. Проблема заключалась в том, что с первыми компьютерами могли работать только хорошо обученные программисты. Чтобы компьютеры стали по-настоящему полезными, им нужно было стать компактными (иначе они не помещались в квартиру или офис) и простыми в обращении.

ПОЛЕТ НА ЛУНУ

В 1969 году люди совершили необычайный подвиг: преодолели более 370 тысяч километров безвоздушного космического пространства и впервые высадились на Луне — с помощью компьютера той же мощности, что и современный карманный калькулятор! Бортовой управляющий компьютер «Аполлона», встроенный в лунный модуль NASA, совершил революцию в компьютерных технологиях. В то время это был один из самых маленьких компьютеров на Земле (и вне ее). NASA и инженеры Массачусетского

ИСТОРИЯ КОМПЬЮТЕРОВ, ЧАСТЬ 2. РАСЦВЕТ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ

ЭНИАК, предок цифровых компьютеров, занимал комнату размером 6 на 12 метров и содержал 18 тысяч вакуумных ламп.
1943–1944 гг.

Язык программирования Кобол (COBOL) — первый язык программирования, приближенный к английскому.
1953 г.

Бортовой управляющий компьютер «Аполлона» отправил компьютерные технологии на Луну.

UNIX — операционная система, разработанная в Лабораториях Белла. С UNIX одновременно могли работать много людей.
1969 г.

1947 г.

Транзистор изобрели ученые Лабораторий Белла. Электронные схемы на транзисторах были меньшего размера и подготовили почву для персональных компьютеров.

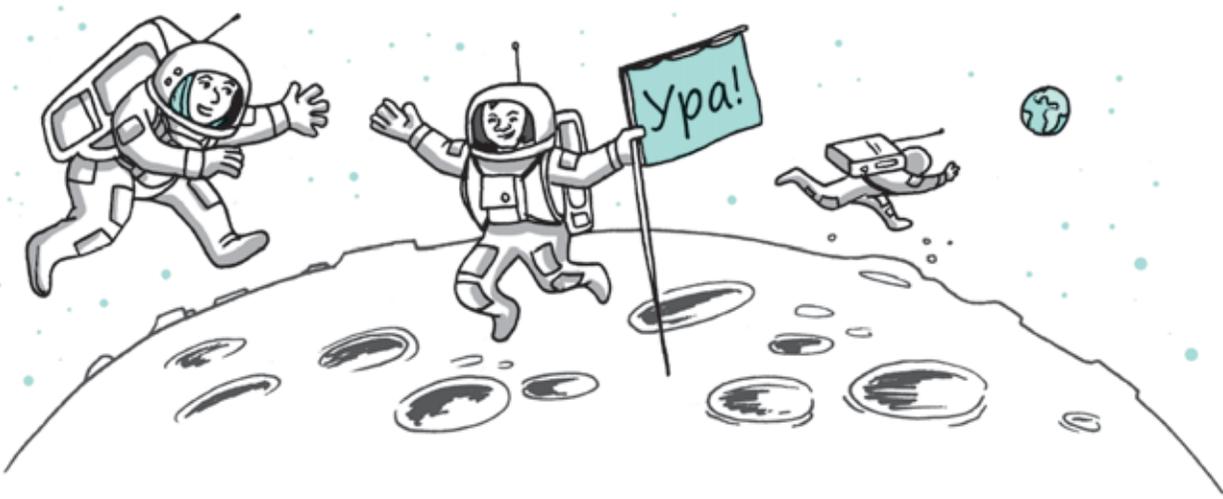


1975 г.

IBM 5100 — первый компьютер современного типа с клавиатурой, дисплеем и встроенной памятью. Microsoft. Билл Гейтс и Пол Аллен основали компанию Microsoft.

ПОЧЕМУ ПРОГРАММИРОВАНИЕ?

технологического института смогли втиснуть компьютер, огромный, как семь холодильников, в устройство размером с микроволновку и весом чуть больше 30 килограммов. Устройство снабдили программой, чтобы космонавты вводили простые команды из глаголов и существительных без помощи программиста. С таким компьютером мог справиться даже пользователь, не умеющий программировать.



Apple. Стив Джобс и Стив Возняк основали компанию и выпустили в продажу свой первый компьютер Apple I.
1976 г.

Персональный компьютер от компании IBM, работал на операционной системе DOS от Microsoft.
1981 г.

Изобретение Всемирной паутины.
1989 г.

1977 г.

Apple II стал первым компьютером массового производства, и пользователи смогли покупать его по доступной цене.



1985 г.

Windows. Компания Microsoft выпустила свою операционную систему. Основан онлайн-сервис America Online.

1988 г.

Первый складной ноутбук UNIX, новинка от компании Compaq.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК



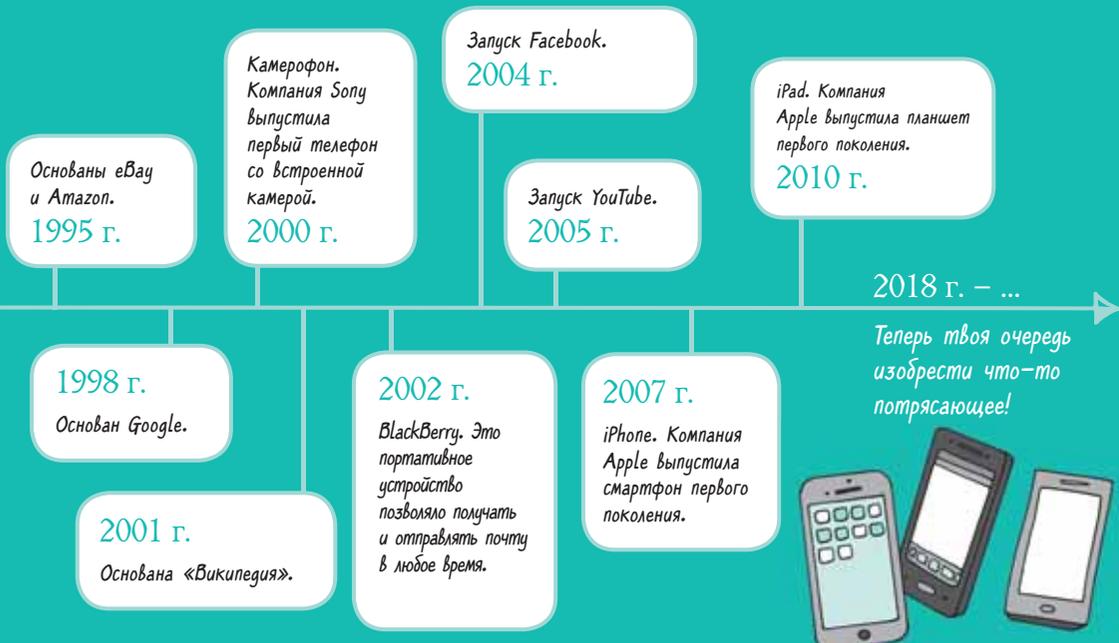
*Один маленький шаг для человека
и гигантский скачок для всего
женского рода!*

МАРГАРЕТ ГАМИЛЬТОН

А ты знаешь, что женщина-математик Маргарет Гамильтон разрабатывала программное обеспечение для двух переносных компьютеров проекта

«Аполлон»? Она придумала не только термин «программная инженерия», но и саму эту работу! В 1960-х годах, когда США взялись за программу полетов на Луну «Аполлон», такой науки, как информатика, не существовало. Не было и программного обеспечения для бортовых компьютеров корабля. Все это изобрели Гамильтон и ее коллеги из Массачусетского технологического института. Их работа обеспечила безопасную высадку человека на Луне и положила начало глобальной индустрии с многомиллиардными оборотами.

ИСТОРИЯ КОМПЬЮТЕРОВ, ЧАСТЬ 3. ЗАРЯ ИНТЕРНЕТА



ПОЧЕМУ ПРОГРАММИРОВАНИЕ?

ЧТО МЫ ИМЕЕМ СЕЙЧАС

Менее чем через десять лет после высадки на Луну первые персональные компьютеры массового производства стали доступными для обычных пользователей. С тех пор благодаря постоянным открытиям в технологиях и компьютерной инженерии наши компьютеры становятся все меньше и быстрее. Беспроводные и мобильные технологии открыли нам мир, где можно в реальном времени поговорить в видеочате с человеком, находящимся на другой стороне земного шара. Мы можем это сделать с помощью маленького устройства, которое легко умещается в кармане.



Я НИКОГДА НЕ СМОТРЕЛА НА КОМПЬЮТЕРЫ С ТАКОЙ СТОРОНЫ. ЗДОРОВО, ЧТО ПРОГРАММИРОВАНИЕ ДАЕТ НАМ ВОЗМОЖНОСТЬ ОТКРЫВАТЬ И ИЗОБРЕТАТЬ ЧТО-ТО НОВОЕ.



ВОТ-ВОТ! Я УЖЕ ХОЧУ УЧИТЬСЯ ПРОГРАММИРОВАТЬ. С ЧЕГО НАЧНЕМ?

Начать можно двумя способами. Можно выбрать язык программирования, пройти обучающие курсы и начать писать программы. Либо изучить, как работают компьютеры, чтобы понять, о чем и как их можно просить. Этим мы и займемся.

Небольшая подсказка: компьютеры, конечно, умные, но бутербродов они делать не умеют.

СИДЕТЬ!

СТОЯТЬ!



КАТИСЬ!



Хороший компьютер!



КАК ОБЩАТЬСЯ СО СВОИМ КОМПЬЮТЕРОМ

ХОТЯ КОМПЬЮТЕРЫ ЧАСТО НАЗЫВАЮТ «ДУМАЮЩИМИ МАШИНАМИ» или «электронным мозгом», сами по себе они мало что могут. Они отлично выполняют указания, но не всегда способны думать самостоятельно. Это значит, что, когда ты просишь компьютер что-то сделать, твои указания должны быть очень точными. Тогда компьютер *сделает все точно так, как ты скажешь.*

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

ЭФФЕКТ АРАХИСОВОГО МАСЛА И ДЖЕМА



Представь, что ты вернулась домой из школы, а на кухне тебя ждет личный робот, готовый выполнять твои приказы. Здорово, правда? Ты проголодалась и просишь робота (давай назовем его СуперБот-3000 или просто Бот) сделать бутерброд с джемом и арахисовым маслом.

— ПРОШУ ДАЛЬНЕЙШИХ УКАЗАНИЙ, — говорит Бот своим электронным робоголосом.

Все, что нужно для бутерброда, уже лежит на кухонной стойке, поэтому ты говоришь:

— Ладно, иди к стойке, возьми хлеб, намажь его джемом и арахисовым маслом и принеси мне.

Бот едет к стойке (я уже говорила, что у него есть колеса?), берет банку с арахисовым маслом, банку с джемом и начинает тереть их о пакет с хлебом.

Компьютеры все понимают *буквально*. Если ты хочешь, чтобы Бот сделал обычный съедобный бутерброд, нужно сказать так:

— Найди хлеб, открой пакет. Возьми два куска. Положи их на доску плашмя, друг рядом с другом. Езжай на полметра вправо. Найди ящик, открой его и возьми из ящика нож. Закрой ящик. Езжай на полметра влево. Найди арахисовое масло. Возьми банку и сними крышку. Погрузи нож в арахисовое масло...



КАК ОБЩАТЬСЯ СО СВОИМ КОМПЬЮТЕРОМ

И так далее.

Чтобы написать программу, нужно знать язык программирования, но еще важнее — понимать, как «думает» компьютер. Только так можно дать компьютеру указания, чтобы он выполнил то, что ты хочешь.

ХМ-М!
ЧТО-ТО СЛОЖНОВАТО.



На самом деле это легко, нужно только разобраться в **ВЫЧИСЛИТЕЛЬНОМ МЫШЛЕНИИ** — то есть научиться планировать, решать задачи и анализировать информацию так, как это делает компьютер. Если разобраться в том, каким образом компьютеры обрабатывают информацию, становится понятно, что может выполнить компьютер, а чего он не может сделать. После этого программировать станет гораздо легче.



А ЧЕМ
ВЫЧИСЛИТЕЛЬНОЕ МЫШЛЕНИЕ
ОТЛИЧАЕТСЯ ОТ ОБЫЧНОГО?

Мы используем свой мозг, чтобы выполнять разные сложные задачи и обрабатывать информацию, которую получаем из окружающего мира. Например, мы находим новые способы решить проблему или сообщаем другим, что мы думаем и чувствуем, с помощью слов и жестов. Компьютеры так не умеют.

Однако они отлично справляются с некоторыми важными умственными задачами.

Во-первых, у них *прекрасная память*. Они помнят всю информацию, какую ты вводила, и с помощью нужной программы способны ее найти. Ты, наверное, тоже

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

многое помнишь — например, когда у твоей лучшей подруги день рождения или как добраться до твоего любимого кафе-мороженого, — но в памяти компьютера могут храниться миллиарды и миллиарды фактов, рисунков, чисел, фотографий, видео, игр и книг.

Во-вторых, компьютеры хорошо выполняют *повторяющиеся* действия, которые составляют значительную часть нашей жизни. Повторяя, то есть делая что-то раз за разом, мы тренируемся и учимся (вспомни, как ты готовишься к тесту или заучиваешь слова для выступления). Компьютеры, правда, от повторения ничему не учатся, но сам процесс у них выходит отлично. Они способны сотни раз повторить вычисление или задачу и при этом не устать, не заскучать и не ошибиться. Например, твой телефон будит тебя по утрам, звонко напоминая, что пора вставать. Будильник — это заложенная в него программа, она повторяется утром каждого учебного дня... даже когда тебе этого совсем не хочется.

Наконец, компьютеры здорово умеют *принимать решения*. Это не важные жизненные решения, какие принимают люди, например «Кем мне стать в будущем?» или «Сказать ли тому человеку, что он мне нравится?». Компьютеры принимают решения, когда нужно ответить на вопрос вроде «Вверх или вниз? Включить или выключить? Шоколад или ваниль?». Правильно настроенные, эти простые решения становятся важной



Этого достаточно, чтобы пересечь Соединенные Штаты поперек более 65 раз!



КАК ОБЩАТЬСЯ СО СВОИМ КОМПЬЮТЕРОМ

частью компьютерной программы. Посмотри, как работает навигатор в машине или смартфоне. Ты вводишь адрес места, куда хочешь отправиться, а навигатор просматривает информацию о доступных маршрутах и решает, отправить тебя налево или направо, по этой дороге или по другой.

Запоминание, повторение и принятие простых решений — вот основы вычислительного мышления. И если ты соберешь их воедино...



**ТЫ ПРОГРАММИРУЕШЬ,
ДЕВОЧКА!**

Но нельзя просто попросить компьютер выполнить эти задачи в произвольном порядке: нужно скомбинировать их так, чтобы компьютеру было понятно. Для этого надо следовать определенным правилам.

Не забывай, как работают компьютеры:

ВВОД ➡➡ ОБРАБОТКА ➡➡ ВЫВОД

Ты уже знаешь, что программировать — значит писать для компьютера указания на языке программирования. Как и в любом другом языке, будь то английский, французский, русский, суахили или хинди, в языке программирования есть правила. Чтобы нас понимали, нужно соблюдать правила. Нельзя расположить слова в произвольном порядке и ждать, что тебя поймут. Книга эта иначе так была бы написана, и, как Йода, говорили бы мы.

Точно так же и с программированием. На самом деле языки программирования гораздо более точные и четкие, чем разговорные языки. Хотя программисты пишут код по-разному, они пользуются одними и теми же основными

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

ДАННЫЕ — это информация, которую ты вводишь в компьютер, чтобы он выполнил задачу или произвел вычисление.

ЛОГИКА — это правила, которые компьютер соблюдает при обработке данных.

элементами. Элементы языка программирования нужно расположить строго по правилам, иначе компьютер не поймет, что ты хочешь².

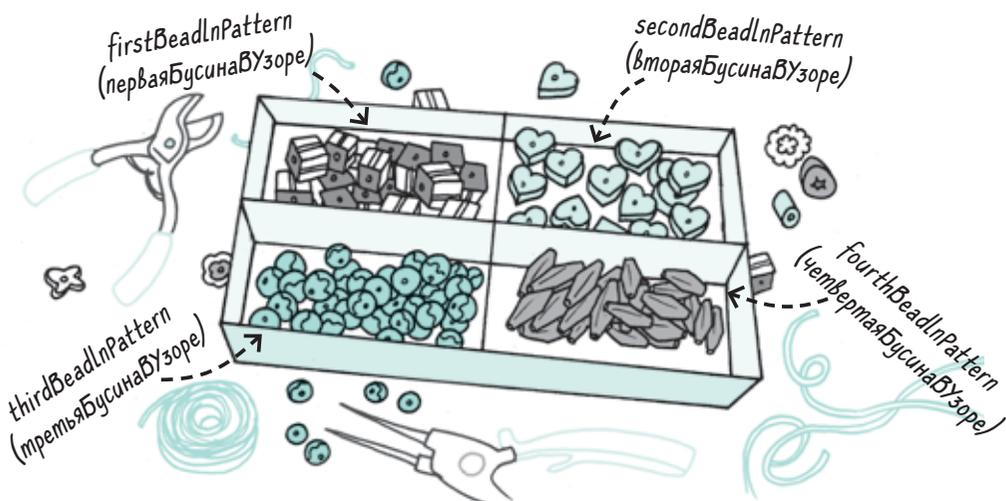
Теперь посмотрим, как с помощью кода попросить компьютер что-нибудь запомнить, повторить или решить.

ПЕРЕМЕННАЯ = ЗАПОМИНАНИЕ

Когда ты пишешь программу, ты даешь компьютеру отдельный кусочек информации, который нужно запомнить. Этот кусочек называется **ПЕРЕМЕННОЙ** (англ. variable). Переменная похожа на коробку, в которой можно хранить информацию.

Ты увлекаешься скрапбукингом, шитьем, рисованием, коллажированием или парикмахерским искусством? Переменные похожи на коробочки, в которых хранятся разные материалы для рукоделия. В жизни ты наклеиваешь на коробки ярлыки, где написано, что лежит внутри: пуговицы, бусины, цветные мелки, заколки для волос — так проще найти то, что тебе нужно. Так же и с переменными. Когда мы присваиваем переменной имя, мы приклеиваем ярлык к «коробке» с информацией. Содержимое может измениться, но коробка

² В этой книге программный код пишется в основном английскими буквами, но для того, чтобы вам было понятно, что он означает, мы приводим его перевод в скобках. — Примеч. ред.



КАК ОБЩАТЬСЯ СО СВОИМ КОМПЬЮТЕРОМ

останется той же. В программировании переменные используют для того, чтобы задавать и хранить разную меняющуюся информацию.

Переменные могут содержать числа, короткие фразы, последовательности чисел (их называют **СТРОКАМИ** (англ. string)) или утверждения. Утверждения могут быть истинными или ложными (англ. true и false) (**ЛОГИЧЕСКИЕ ЗНАЧЕНИЯ**).

Вот примеры переменных и информации в них:

```
имяПеременной = значение;
```

Числа: переменные могут содержать числовые данные. В следующем коде переменные `currentAge` (мойВозраст), `costOfIceCream` (ценаМороженого), `daysLeftOfSchool` (днейОсталосьВУчебномГоду), `stringLengthInches` (сантиметровВСтроке) — числовые.

```
currentAge = 12;           daysLeftOfSchool = 234;
costOfIceCream = 30,5;    stringLengthInches = 17;
```

Текст: переменные могут содержать длинные строки текста. Текстовые строки всегда заключаются в кавычки. Зададим строки `mentorsName` (имяСтаршейПомощницы), `favoriteSong` (любимаяПесня), `dayOfTheWeek` (деньНедели) и `statusUpdate` (обновлениеСтатуса).

```
mentorsName = "Гуля";
favoriteSong = "Don't Stop Believin'";
dayOfTheWeek = "Пятница";
statusUpdate = "Наконец-то учусь программировать!";
```

Логические значения: можно присваивать переменным значения `true` («истина») или `false` («ложь»). Они используются, если нужно принять решение. Добавим логические переменные `isWeekend` (сейчасВыходные), `loggedIn` (вошлаВАккаунт), `brotherIsAwake` (братПроснулся) и `stillHaveString` (осталасьЕщеСтрока).

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

```
isWeekend = true;
loggedIn = true;
```

```
brotherIsAwake = false;
stillHaveString = true;
```

ЦИКЛ = ПОВТОРЕНИЕ

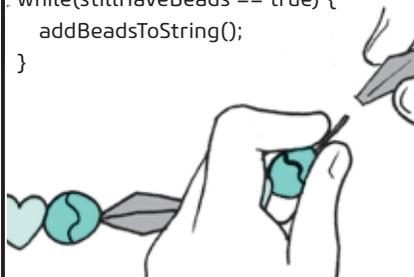
ЦИКЛ (англ. loop) — это элемент кода, который говорит компьютеру, что нужно что-то повторить. Так же как твой будильник звонит по утрам в одно и то же время, цикл — это конструкция в коде, которая повторяет процесс.

Предположим, у тебя есть коробка с бусинами и ты нанизываешь их на нитку. Ты хочешь расположить бусины в определенном порядке: ■ ♥ ● ◆ и снова ■ ♥ ● ◆. Повторение одного и того же узора — это и есть цикл.

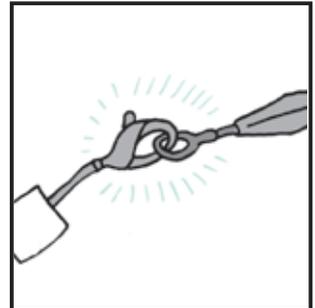


ШАГ ПЕРВЫЙ:
ПРИДУМАЙ УЗОР.

```
.while(stillHaveBeads == true) {
  addBeadsToString();
}
```

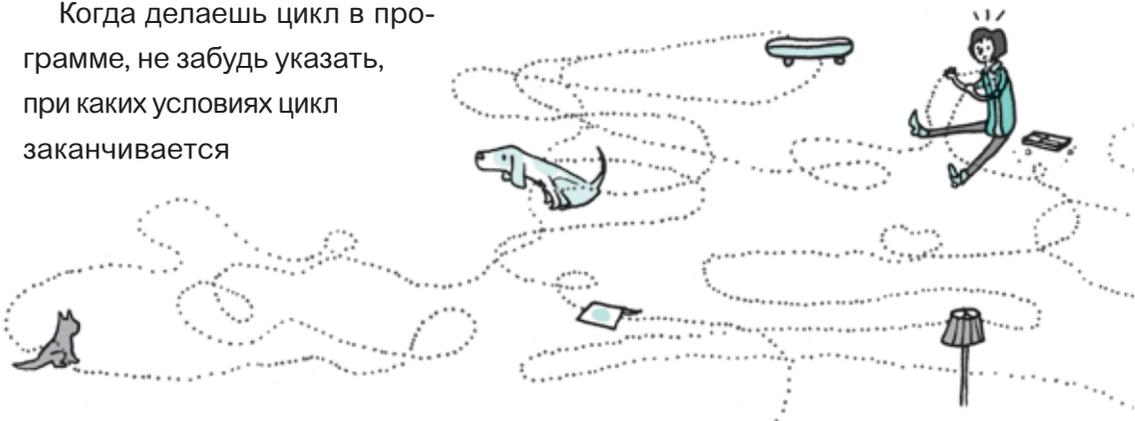


ШАГ ВТОРОЙ: ПОВТОРЯЙ
УЗОР, ПОКА БУСИНЫ
НЕ ЗАКОНЧАТСЯ.



ШАГ ТРЕТИЙ:
В КОНЦЕ ПРИКРЕПИ
ЗАСТЕЖКУ.

Когда делаешь цикл в программе, не забудь указать, при каких условиях цикл заканчивается



КАК ОБЩАТЬСЯ СО СВОИМ КОМПЬЮТЕРОМ

или сколько нужно повторов. Иначе получится бесконечный цикл — в нашем примере это будет самое длинное на свете ожерелье!

УСЛОВНАЯ КОНСТРУКЦИЯ = ПРИНЯТИЕ РЕШЕНИЙ

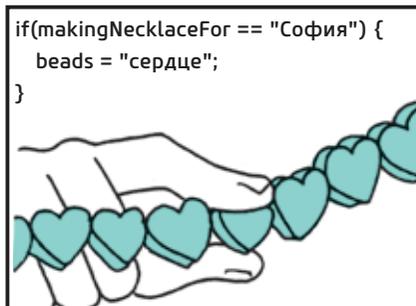
УСЛОВНЫЕ КОНСТРУКЦИИ используются, когда тебе нужно, чтобы компьютер принял какое-то решение. Ты, скорее всего, уже сталкивалась с такими конструкциями: наверняка родители или дедушка с бабушкой говорили тебе что-нибудь вроде «Ты можешь посмотреть этот фильм при условии, что сначала сделаешь все уроки». То есть конструкция ставит условие для того, чтобы сделать или решить что-то.

В программировании условные конструкции обычно формулируются с помощью слова «если» (англ. if), которое мы постоянно используем в обычной жизни, чтобы принимать решения. *Если* пойдет дождь, *когда* я соберусь уходить из дома, *тогда* (англ. then) я надену резиновые сапоги. *Если* я проголодаюсь, *когда* приду домой из школы, *тогда* я чем-нибудь перекушу. Теперь попробуй сама вспомнить какую-нибудь условную конструкцию из жизни:

ЕСЛИ _____, **ТОГДА** _____.



ЕСЛИ Я ДЕЛАЮ
ОЖЕРЕЛЬЕ ДЛЯ СОФИИ...



```
if(makingNecklaceFor == "София") {  
    beads = "сердце";  
}
```

ТО Я БЕРУ БУСИНЫ В ФОРМЕ
СЕРДЦА...

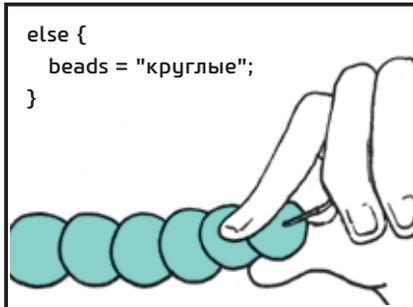


ПОТОМУ ЧТО ЭТО
ЕЕ ЛЮБИМЫЕ.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК



ЕСЛИ Я ДЕЛАЮ ОЖЕРЕЛЬЕ
ДЛЯ КОГО-ТО, КРОМЕ
СОФИИ...



ТО Я БЕРУ КРУГЛЫЕ
БУСИНЫ...



ПОТОМУ ЧТО ОНИ
НРАВЯТСЯ ВСЕМ.

Но иногда что-то должно произойти только тогда, когда утверждение ложно. Такое условие обозначают словом «иначе» (англ. else). Если пойдет дождь, когда я соберусь уходить из дома, тогда я надену резиновые сапоги. Иначе (в противном случае) я надену теннисные туфли. Если я проголодаюсь, когда приду домой из школы, тогда я чем-нибудь перекушу; иначе сразу сяду за уроки. Слова «если» и «иначе» задают условия.

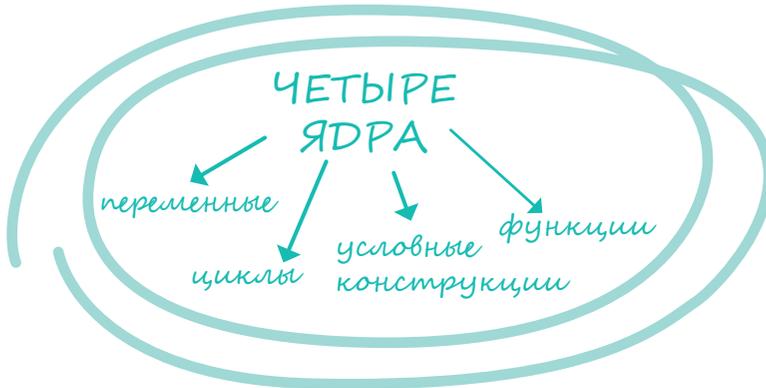


В языках программирования используются английские слова. Запомни: конструкция «если-тогда-иначе» в настоящей программе выглядит как if-then-else.

Переменные, циклы и условные конструкции — это базовые «кирпичики».

Из них можно составлять программы и сообщать компьютеру, что он должен делать. С этих элементов начинается создание компьютерной игры или сайта социальной сети. Они настолько важны, что вместе с функциями (о них ты узнаешь

КАК ОБЩАТЬСЯ СО СВОИМ КОМПЬЮТЕРОМ



в следующей главе) мы зовем их **ЧЕТЫРЬМА ЯДРАМИ** информатики. Используя «кирпичики», ты сможешь составить схему ситуаций из жизни и запрограммировать их по правилам, понятным для компьютера.

Теперь ты можешь начать творить.

ЗАПОМИНАНИЕ, ПОВТОРЕНИЕ И ПРИНЯТИЕ РЕШЕНИЙ — ЭТО ВСЕ, ЧТО НУЖНО ДЛЯ РАБОТЫ МОЕЙ ЭЛЕКТРОННОЙ ПОЧТЫ ИЛИ ИГРЫ НА ТЕЛЕФОНЕ?



Ну на самом деле все чуточку сложнее, но это — основа любого кода, включая тот, на котором работают крупные сервисы вроде «Википедии», Instagram или WhatsApp. Об этом мы расскажем в следующей главе. А еще там будут тако³!

*Р. С. Почему курица опять перешла через дорогу?
Потому что ее запрограммировали на бесконечный цикл!*

³ Тако (исп. taco) — блюдо мексиканской кухни. Представляет собой сандвич с разнообразной начинкой: жареным мясом, овощами, зеленым салатом, фасолью. — *Примеч. пер.*

МАФФИНЫ С АРАХИСОВЫМ МАСЛОМ

ИНГРЕДИЕНТЫ:

- 240 г муки, 1 ст л разрыхлителя, 240 мл молока,
- 2 яйца, 90 г сахара,
- 120 г арахисового масла,
- 1 ч л соли

1. Разогрейте духовку до 200 °С.
2. Смешайте муку с разрыхлителем.
3. Смешайте молоко, масло, сахар с арахисовым маслом и добавьте в муку.
4. Разлейте смесь по формам.
5. Выпекайте 20 минут.



Как пройти на СТЕП-киносеанс Люси

1. Выйди из школы
2. От дуза налево.
3. От почты направо.
4. На круговом перекрестке направо.
5. От леса направо.
6. От клена направо.
7. От дома Уиверов налево.
8. От дома Браунов налево.
9. Налево — и ты на месте!



18:00 — начало
19:00 — фильм 1
21:00 — фильм 2
21:30 — фильм
* мороженое
23:00 — фильм 3
НОЧЕВКА!



СОБИРАЕМ ВСЕ ВМЕСТЕ

ЕСЛИ ТЫ УЖЕ СЛЫШАЛА ЧТО-ТО О ПРОГРАММИРОВАНИИ ПРЕЖДЕ, ЧЕМ браться за эту книгу, тебе наверняка знакомо слово **«АЛГОРИТМ»**. Это длинное, таинственно звучащее слово — та самая магическая формула, благодаря которой работают поисковые системы и интернет-магазины предлагают тебе товары по вкусу. Алгоритмы есть не только в компьютерах. Ты пользуешься ими каждый день и, наверное, даже не замечаешь этого.

Алгоритм — это цепочка указаний, которые необходимо выполнить в определенном порядке, чтобы справиться с задачей. Рецепт

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

маффинов, порядок движений в танце или указания, как пройти к дому подруги, — все это алгоритмы. Даже такой привычный процесс, как твои утренние сборы, можно рассматривать как алгоритм. Ты просыпаешься, встаешь с постели, чистишь зубы, выбираешь наряд, одеваешься — все для того, чтобы надеть куртку, ботинки и выйти из дома. Ты выполняешь этот алгоритм почти каждый день, чтобы получить один и тот же результат: привести себя в порядок и отправиться в школу.

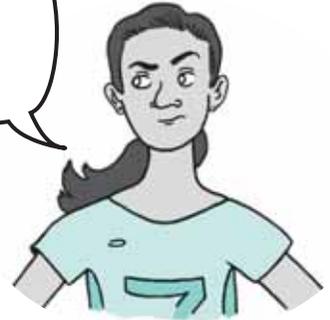
Кажется, это очень простые действия, настолько простые, что ты можешь собраться, даже толком не проснувшись. Но на самом деле каждое из этих действий можно разделить на множество простых этапов. Даже надевая носки, ботинки и завязывая шнурки, ты выполняешь цепочку из нескольких шагов.



СОБИРАЕМ ВСЕ ВМЕСТЕ

Помнишь нашего друга Бота из прошлой главы? Когда он готовил бутерброд с джемом и арахисовым маслом, мы увидели, что составить алгоритм не так просто. Ведь любую процедуру приходится описывать подробно, в мельчайших деталях. Алгоритм — это четкая инструкция. Ее может выполнять кто угодно, в том числе и компьютер, который все понимает буквально.

ТО ЕСТЬ КОГДА Я РАЗМИНАЮСЬ
ПЕРЕД ПРОБЕЖКОЙ, ОБУВАЮСЬ
ИЛИ МОЮ ГОЛОВУ,
Я ВСЕ ЭТО ВРЕМЯ СЛЕДЮЮ
АЛГОРИТМАМ?



И это еще не все. Каждый день ты используешь одни из самых подробных программных алгоритмов в мире. Когда ты ищешь что-то в Google, проверяешь прогноз погоды на смартфоне или просматриваешь персональные рекомендации в видеосервисах вроде YouTube и Netflix или в интернет-магазинах, всю эту информацию находят и предоставляют тебе сложные алгоритмы, разработанные и написанные программистами.

Если вам понравился «Мой сосед Тоторо», вам могут также понравиться:



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

ТО ЕСТЬ ПРОГРАММА —
ЭТО БОЛЬШОЙ АЛГОРИТМ,
НАПИСАННЫЙ ТАК, ЧТОБЫ
КОМПЬЮТЕР МОГ ЕГО
ПОНЯТЬ.



Совершенно верно. Алгоритм в программе — это последовательность инструкций, которую компьютер должен выполнить. Но алгоритм — это *еще не код*. Алгоритм — это только этапы задачи. Их еще предстоит *превратить* в код.

Ты уже знаешь, что компьютеры отлично запоминают информацию, повторяют действия и принимают простые решения. В алгоритме все простые действия приведены в том порядке, в котором их должен выполнить компьютер, чтобы получить результат. Когда алгоритм составлен, его нужно перевести в код — на том языке программирования, который лучше подходит для выбранной задачи.

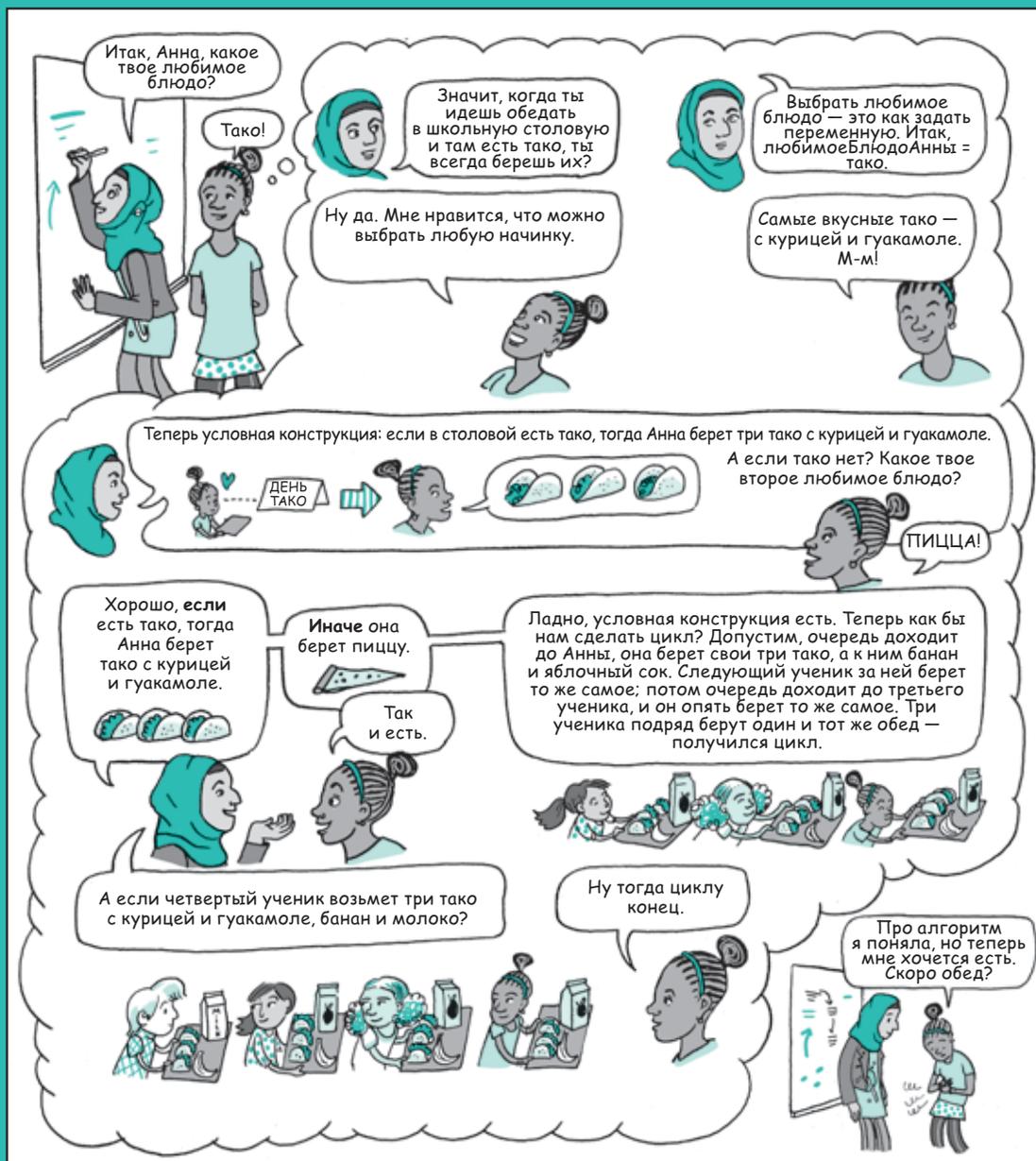
Возьмем для примера что-нибудь, что ты делаешь каждый день, и распишем это в форме алгоритма, где будут переменные, циклы и условные конструкции.

Есть идеи?



МОЖЕТ БЫТЬ, «Я ДУМАЮ,
ЧТО ВЗЯТЬ НА ОБЕД»?

Отлично. Составим «обеденный алгоритм». Гуля, выходи к доске.



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Это всего лишь один простенький алгоритм, но при желании такой можно сделать для любой из вас на весь учебный год с самыми разными блюдами. Каким бы большим и сложным ни получился твой алгоритм, основные шаги останутся такими же.

Давай попробуем записать наш обеденный алгоритм, используя **ПСЕВДОКОД**, то есть напишем очень точную инструкцию обычным человеческим языком, прежде чем переводить ее в программный код.

если Анна берет тако:

пока есть пустые тако:

работник столовой наполняет тако начинкой

иначе:

работник столовой дает Анне пиццу

работник столовой дает Анне
яблочный сок и бананы

Я МОГУ ЭТО ПРОЧИТАТЬ.
Я ВСЕ ПОНИМАЮ! И ВИЖУ
КАЖДЫЙ ШАГ.



Этим и хорош псевдокод. Можно сначала набросать программу на своем родном языке, а потом перевести ее в компьютерный код. Давай посмотрим, как нам это сделать.

НЕ ПОВТОРЯЙСЯ

Некоторые элементы псевдокода повторяются. В нашем примере тако всегда будут с начинкой «курица и гуакамоле». Когда ты пишешь программу, повторяющиеся элементы встречаются очень часто, потому что тебе то и дело приходится просить компьютер сделать что-то несколько раз.

Но писать все заново не нужно! Первым делом программисты учат принцип **«НЕ ПОВТОРЯЙСЯ»**. Хороший алгоритм (и хороший код) должен быть ясным, как научил нас наш друг Бот, и эффективным. Пиши просто, понятно и не перегружай программу повторяющимися элементами. К счастью, для этого есть несколько хитрых приемов.

ОБЕДЕННЫЙ ЦИКЛ

Давай представим, что вторник у нас — день тако (ура!). Каждый раз, когда Анна берет тако, она заказывает одну и ту же начинку, поэтому указания в нашем алгоритме — и в коде — будут повторяться: начинить тако курицей и гуакамоле. Код будет выглядеть примерно так:

```
taco1.fill("курица")
```

```
taco1.fill("гуакамоле")
```

```
taco2.fill("курица")
```

```
taco2.fill("гуакамоле")
```

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

```
taco3.fill("курица")
```

```
taco3.fill("гуакамоле")
```

Ничего себе, сколько строчек печатать! Вместо того чтобы прописывать одни и те же инструкции для каждого тако, можно написать всего один цикл.

```
for each taco in annaOrder:
```

```
  taco.fill("курица")
```

```
  taco.fill("гуакамоле")
```

Теперь во всех тако будет одинаковая начинка, а в нашем коде — всего три строчки. Но что, если мы хотим тако с разной начинкой? Тогда нам пригодятся функции.

ФУНКЦИИ

ФУНКЦИИ (англ. functions) — последнее из четырех ядер информатики (наряду с переменными, циклами и условными конструкциями). Это кусочки кода, которые в программе выполняют определенную работу. Функции могут служить разным целям, но самая важная из них — избавить тебя от необходимости



СОБИРАЕМ ВСЕ ВМЕСТЕ

снова и снова писать один и тот же код. Достаточно написать функцию и дать ей название. Всякий раз, когда тебе понадобится функция, ты сможешь запустить ее, набрав ее название. После этого функция выполнит свою работу в этой части программы. Это все равно что создать ярлык или кнопку, которая автоматически выполняет нужные действия, так что тебе не приходится каждый раз прописывать указания заново.



Наверняка какой-нибудь ученый уже трудится над роботом — сборщиком одежды. Вернемся к нашему обеденному алгоритму и посмотрим, как функция работает внутри программы.

ЗАДАЕМ ФУНКЦИЮ

Любую функцию первым делом нужно задать. Придумай ей название — иначе твоя программа не сможет найти этот кусочек кода, когда он нам понадобится. Название должно быть понятным и содержательным. Назовем нашу обеденную функцию `fill_my_taco` (начинка_моего_тако)!

Название есть? Теперь пора писать код для функции — в частности, задать ее **ПАРАМЕТРЫ**. Параметры — это маленькие переменные внутри функции. Они содержат информацию, которая может изменяться.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Код будет выглядеть примерно так:

```
def fill_my_taco(taco, item1, item2):

    taco.fill(item1)

    taco.fill(item2)
```

`def` — сокращение от англ. `define` — «определить, задать». Используется для объявления функций. Встречается в словаре терминов.

Параметры указаны в скобках после названия функции. В функции под названием `fill_my_taco` мы можем положить в `taco` две разные начинки.



ВЫЗЫВАЮ ФУНКЦИЮ!

Как только функция готова, ее остается лишь применить. Это важный момент: недостаточно просто назвать функцию и написать для нее код. Обязательно нужно указать компьютеру, когда и где использовать функцию. Для этого нужно напечатать название функции в том месте программы, где она должна сработать, то есть «вызвать» функцию.

Код для вызова функции будет выглядеть примерно так:

```
fill_my_taco(taco1, "курица", "гуакамоле")
```

Для других так можно выбрать какую-нибудь другую начинку, например:

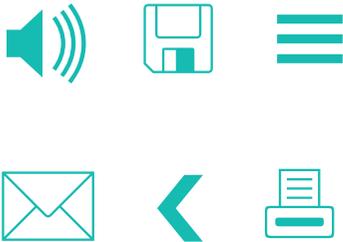
```
fill_my_taco(taco2, "тофу", "сальса")
```

СОБИРАЕМ ВСЕ ВМЕСТЕ

```
fill_my_taco(taco3, "фасоль", "овоци")
```

И вот у нас три разных тако. М-м... разнообразие!

Для пользователя, который смотрит на экран, функции в программе могут выглядеть как кнопки или ярлыки. Например, кнопка «Печать» в текстовом редакторе или кнопка «Назад» в браузере. Когда ты пользуешься программой и нажимаешь такую кнопку, она вызывает из кода функцию и запускает ее.



БИБЛИОТЕКИ

Функции отлично помогают сэкономить время при написании программ, но для этого есть и другие хитрости, например библиотеки. Мы говорим сейчас не про твою районную библиотеку, куда ты ходишь за книгами. (Мы ♥ библиотеки!) В библиотеках кода хранятся полезные алгоритмы и уже написанные фрагменты кода. Другие люди могут использовать готовые фрагменты в собственных программах. В таких библиотеках можно отыскать различные виды кода.

- ★ Поисковые алгоритмы. На таких алгоритмах работают поисковики Google или «Яндекс».

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

- ★ Алгоритмы сортировки. С их помощью можно упорядочить данные по алфавиту, по номеру, названию и т. д.
- ★ Алгоритмы выдачи рекомендаций. На основе введенной информации или данных поиска такой алгоритм формирует предложения для конкретного пользователя (например, «фильмы для вас» на Netflix или товары на Amazon).

Использование готового кода, такого как библиотеки и функции, не только экономит программистам время и силы, но и позволяет делиться с другими программистами своими красивыми, точными и эффективными фрагментами кода.



СОБИРАЕМ ВСЕ ВМЕСТЕ

ПРОГРАММНЫЙ ИНТЕРФЕЙС ПРИЛОЖЕНИЯ

Другой полезный инструмент программиста — это **ПРОГРАММНЫЙ ИНТЕРФЕЙС ПРИЛОЖЕНИЯ**. Благодаря интерфейсу **ПРИЛОЖЕНИЯ** могут обмениваться информацией. Сейчас такие интерфейсы есть у большинства приложений. Например, у Gismeteo, карт Google, Instaglam и Twitter. С их помощью можно встроить на сайт твиты или отобразить информацию о погоде в том месте, где находится пользователь. Программные интерфейсы создают окошко в коде, чтобы другие программы могли поделиться своими данными. Это отличный пример того, как совместная работа программистов порождает удивительные технологии.

Три участницы «Программирования для девочек» разработали замечательное приложение. В нем девочки использовали алгоритмы, программные интерфейсы и функции.



Приложение помогает выбрать прическу по погоде. Его создательницы, Кениша Дж., Серена В. и Фейт У., так описали свою идею: «Мы заметили, что общая проблема всех женщин в том, что мы не просыпаемся сразу с идеальной прической. Когда мы собираемся, нам приходится

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК



смотреть, какая на улице погода, какая влажность, не пойдет ли дождь? В результате мы тратим драгоценное время на то, чтобы подобрать новую несложную прическу и спокойно выйти на улицу. Чтобы решить эту проблему, мы придумали „Зеркало погоды“.

Это мобильное приложение, где нужно выбрать свою длину и тип волос и ввести почтовый индекс места, куда вы пойдете. На основе этой информации приложение подберет для вас прическу. Больше не нужно беспокоиться о том, что погода нарушит ваши планы. Прежде чем выйти из дома, не забудьте заглянуть в „Зеркало погоды“!»

Девочки написали приложение на языке JavaScript. Давай посмотрим на фрагмент их кода. Если не все понятно, читай комментарии к коду. В них подробно описан алгоритм работы «Зеркала погоды».



```
var zip = document.getElementById("usersZip").value;
```

```
$.ajax({
```

```
  url: "https://api.wunderground.com/
  api/2f800aa485a60839/geolookup/
  conditions/q/" + zip + ".json",
```

```
  dataType: "jsonp",
```

Эта строчка проверяет, какой почтовый индекс ввела пользовательница, и сохраняет его в переменную zip (от англ. zip code — «почтовый индекс»).

Эта строчка ищет последние данные о погоде по введенному индексу через программный интерфейс сайта Weather Underground.

СОБИРАЕМ ВСЕ ВМЕСТЕ

```

    success : function(weatherData) {
      var location = weatherData['location']['city'];
      var temp_f = weatherData['current_observation']
        ['temp_f'];
    }
  });

  alert("Current temperature in " + location + " is: " + temp_f);

```

Если интерфейс работает, эта строчка берет с сайта информацию о погоде и сохраняет ее.

Эта строчка выбирает из данных о погоде название города и сохраняет его в переменной location.

Эта строчка выбирает из данных о погоде температуру и сохраняет в переменной temp_f (f означает «градусы Фаренгейта»⁴).

Девочки используют собранную информацию, чтобы сообщить пользовательнице, какая сейчас погода в ее городе.

Как Кениша, Серена и Фейт создали это приложение? Они задали переменные и соединили их в простой, удобный алгоритм. Потом написали код, добавили функции и программный интерфейс, собирающий информацию о погоде. Так получилось приложение, с которым ты в любую погоду будешь выглядеть стильно и изящно, и твоя прическа будет в порядке!

Поэтому если...

...ты уже готова создать что-то свое, тогда загляни в следующую главу. В ней мы расскажем, с чего начать.

Если ты еще не знаешь, что именно хочешь создать, тогда попробуй перелистнуть до седьмой главы. Возможно, примеры в ней вдохновят тебя на что-то интересное, что можно воплотить в жизнь средствами программирования!

Но сначала давай выясним, кто из животных больше на тебя похож: щенок или котенок.

⁴ Шкала Фаренгейта используется для обозначения температуры в США. В России применяется шкала Цельсия. — Примеч. пер.





С ЧЕГО НАЧАТЬ?

НУ, КТО ГОТОВ НАЧАТЬ СВОЙ ПРОЕКТ?

Что ты хочешь создать? Как будешь это делать? Какой у тебя план?

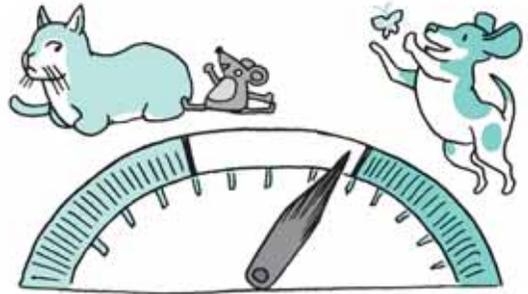
Если эти вопросы застигли тебя врасплох, не волнуйся: так бывает со всеми. Начинать — это самое сложное в любом деле. Однако начало может стать самой интересной и важной частью работы. Начинать новое дело можно по-разному.

НАЧНЕМ СНАЧАЛА

Подход к новому делу у каждого свой, но обычно он находится на шкале между двумя точками. Я называю их «щенок на прогулке» и «котенок с игрушкой».

«Щенки на прогулке» обожают новое.

Начинания их не пугают: они фонтируют идеями. Щенки полны вдохновения и энтузиазма и бросаются к любой новой идее. Они обладают невероятной энергией, но порой рискуют потеряться, погнавшись за бабочками новых идей, и выдохнуться еще до того, как работа будет завершена.



«Котята с игрушками» ведут себя совершенно иначе. Иногда кажется, что новое дело их не очень-то интересует. Сама мысль о том, что пора за него браться, может их слегка ошеломить. Но в то же время они оценивают начинание, обдумывают, планируют и смотрят, как к нему лучше подобраться. И когда они наконец вонзают когти в игрушку, начинается настоящее веселье.

В идеале хорошо оставаться где-то посередине: ты не боишься приступить к чему-то новому, но вместе с тем обдумываешь, чего именно ты хочешь и как этого достичь.

Хорошая новость в том, что неважно, «щенок» ты или «котенок» и насколько амбициозен твой проект. Потому что существует проверенный способ, как сделать первый шаг.

Есть идеи, что это за способ?

Справиться с большим делом легче всего, если разбить его на более мелкие и посильные задачи. Так у тебя всегда будет ясная цель, и объем работы давить не будет.

С ЧЕГО НАЧАТЬ?

НУ, КОГДА МНЕ ПРЕДСТОИТ БОЛЬШОЕ ДЕЛО, НАПРИМЕР УБРАТЬ В КОМНАТЕ, Я РАЗБИВАЮ ЕГО НА МЕЛКИЕ ЗАДАЧИ И ВЫПОЛНЯЮ ИХ ПО ОЧЕРЕДИ.



ТОГДА КАЖЕТСЯ, ЧТО РАБОТЫ МЕНЬШЕ, И Я ТОЧНО ЗНАЮ, ЧТО НУЖНО СДЕЛАТЬ.



Посмотрим, как это применить к проекту по программированию.

РАЗРАБОТКА — СОЗДАНИЕ — ТЕСТИРОВАНИЕ

При проектировании нового продукта (не только в программировании) дизайнеры и программисты разделяют работу на этапы и действуют по циклу «разработка — создание — тестирование». Выглядит это так.

Сначала ты разрабатываешь свой продукт, потом воплощаешь его в реальность и тестируешь, чтобы убедиться, что он делает именно то, что ты хотела. Бывает,



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК



что тестирование возвращает тебя на стадию разработки. Так происходит, когда что-то не работает так, как надо, или нуждается в улучшении.

А ЕСЛИ Я НЕ ЗНАЮ,
ЧТО РАЗРАБАТЫВАТЬ?

Для этого тоже есть своя процедура. Обычно программирование начинается с нее, и для большинства это самая интересная часть!

ИДЕМ НА МОЗГОВОЙ ШТУРМ!

Прежде чем написать хотя бы одну строчку кода, стоит определиться, какую проблему ты хочешь решить и что тебе в этом поможет. Может быть, у тебя есть идея, замысел какого-нибудь приложения или устройства, которое ты хотела бы иметь. Или ты уже пользуешься каким-то устройством и мечтаешь его усовершенствовать. Возможно, у тебя еще нет идей, но есть представление о том, как твое будущее творение может работать или выглядеть. Толчок вдохновению может дать спектакль, стихотворение, музыкальное произведение, общественная проблема, вид спорта или даже любимый оттенок синего. Все это отличные отправные точки для **МОЗГОВОГО ШТУРМА** — свободного, ничем не ограниченного размышления, способного породить идею для твоего проекта.

С ЧЕГО НАЧАТЬ?

Я ЗНАЮ, ЧТО ТАКОЕ МОЗГОВОЙ ШТУРМ, НО С ЧЕГО НАЧИНАТЬ? О ЧЕМ ДУМАТЬ?



Проще всего взять интересную для тебя тему. Попробуй представить ситуацию, где требуется решение, например: «Когда я засекаю время для пробежки, было бы неплохо _____».

«Когда я ищу информацию о вымирающих животных, хотелось бы знать _____».

«Пока я смотрю в Интернете, как украсить торт, разве не здорово было бы _____?»

«Если бы я могла выбрать любое приложение, я выбрала бы _____».

«Я пользуюсь этим устройством каждый день, но могла бы улучшить его, если _____».

Цель мозгового штурма — использовать воображение и начать придумывать идеи. Поэтому посмотри в Интернете, сходи в библиотеку, пройди по музеям, полистай журналы, посмотри рекламные объявления, погуляй по улицам — взглядишь в окружающий мир, вслушайся в него, принюхайся, потрогай, попробуй его на вкус. Найди интересные истории и посмотри, что можно придумать.

ПОХОЖЕ НА КОЛЛАЖИ.
ЛЮБЛЮ ДЕЛАТЬ КОЛЛАЖИ!



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Именно. Вдохновение можно найти где угодно, если усердно искать. Мозг видит закономерности и связи в совершенно неожиданных вещах, поэтому можно немного расслабиться и дать мыслям течь свободно. «Щенкам» самое время погоняться за бабочками, посмотреть, куда они приведут. А «котята» в процессе мозгового штурма смогут повеселиться и размять лапы, не прилагая чрезмерных усилий.

СДЕЛАЙ ЭТО КРАСИВО

Белые и пробковые доски, разноцветные листки для заметок, кнопки, клейкие листки и стикеры. Текстовые делители, маркеры, цветные мелки и карандаши. Ежедневники, папки, блокноты и цветной картон. Страницы из журналов, открытки, фотографии, шаблоны проектов, образцы цвета, постеры к фильмам, обложки.

Цитаты, строчки из песен, высказывания, символы. Все это пригодится тебе для настоящего мозгового штурма. Повесь на стену доску вдохновения, запиши идеи на листочках и прикрепи их к доске. Придумай броский заголовок или название для своего изобретения и напиши его на стене (только убедись, что взрослые не возражают!). Нарисуй главного героя



С ЧЕГО НАЧАТЬ?

своей игры или логотип приложения. Запиши самые важные вопросы и прикрепи их наверху доски. Запиши все ответы, какие только сможешь придумать, и выдели их разными цветами. Наглядное представление помогает думать.

С ПОМОЩЬЮ ДРУЗЕЙ

А знаешь, какое самое важное, сверхсекретное оружие для мозгового штурма?

ТВОИ ДРУЗЬЯ! Ключ к мозговому штурму — это **СОТРУДНИЧЕСТВО**, или совместная работа. Обмениваться идеями — это не только весело, но и полезно. Мы все смотрим на вещи по-разному, и другой человек может предложить что-то такое, что тебе не приходило в голову. Сотрудничество помогает развивать идеи и рассмотреть проблему со всех сторон. Не забывай: цель мозгового штурма — придумать как можно больше идей. Не нужно стесняться или беспокоиться, что твоя идея кому-то не понравится. Не нужно останавливаться и оценивать предложения других. Пусть мысль течет свободно — решить, какие идеи пригодятся, можно будет позже. Поэтому очень важно внимательно относиться ко всем предложениям и сделать так, чтобы каждый свободно выражал свои мысли и не боялся критики.



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

РАЗРАБОТКА

Когда мозговой штурм завершится и идея начнет принимать четкие очертания, нужно расставить приоритеты, то есть убедиться, что это именно та идея, которую ты:

- а) хочешь воплотить,
- б) можешь воплотить
- и
- в) считаешь достойной воплощения.

Чтобы понять, отвечает ли идея всем этим пунктам, нужно задать себе несколько важных вопросов. Поэтому давай сыграем в игру. Она называется...

ШЛИФОВКА БУДУЩЕГО
ИЗОБРЕТЕНИЯ

Думать, что и как разрабатывать, — все равно что принимать решения в ролевой игре. Пусть первый уровень нашей игры будет таким:

> НАЧАЛО



УРОВЕНЬ 1:
НУЖЕН ЛИ НАМ ЭТОТ ПРОДУКТ?



Подумай, чего ты хочешь. Это товар или услуга? Оно нужно для развлечения или может приносить пользу? Возможно, это приложение будет напоминать школьнику, когда нужно сдать домашнее задание? Или игра, которая учит подростков снимать наушники, когда переходишь дорогу? А может,

С ЧЕГО НАЧАТЬ?

это новый фильтр для фотографий или видео? Постарайся найти то, что нужно людям, и подумай, как твое будущее изобретение им поможет.

ОТВЕТ: ДА.
ПОЗДРАВЛЯЮ!
ОТЛИЧНАЯ ИДЕЯ!
> ДАЛЕЕ

ОТВЕТ: НЕ УВЕРЕНА.
ПРОДОЛЖАЙ
МОЗГОВОЙ ШТУРМ.
> НАЧАТЬ ЗАНОВО

УРОВЕНЬ 2:
ОН УЖЕ СУЩЕСТВУЕТ?



Как узнать, существует ли уже что-то подобное тому, что мы задумали? Провести исследование! Поищи аналогичные проекты в Интернете, в магазинах, везде, где такое можно достать. Убедись, что никто еще не сделал того, что задумала ты.

ОТВЕТ: НЕТ.
ПОЗДРАВЛЯЮ! НОВЫЙ
ПОЛЕЗНЫЙ ПРОДУКТ!
> ДАЛЕЕ



ОТВЕТ: ДА.
МОЖНО ЛИ ЕГО УЛУЧШИТЬ?
ПРОДОЛЖАЙ
ИССЛЕДОВАНИЕ.
> НАЧАТЬ ЗАНОВО

В этом случае попробуй взглянуть на уже существующие продукты: нельзя ли их как-то улучшить? Почитай отзывы, поговори с родными и друзьями о том, что им хотелось бы видеть в продуктах, которыми они пользуются. Может быть, изделие стоит лучше приспособить для конкретных пользователей, например детей или подростков. Помни: даже в условиях конкуренции ты всегда можешь найти подходящую потребность и удовлетворить ее.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК



УРОВЕНЬ 3: БУДУТ ЛИ ИМ ПОЛЬЗОВАТЬСЯ?

Подумай, будет ли спрос на твою идею. Заинтересуются ли люди твоим изобретением? Будут ли они им пользоваться? Понравится ли оно им? Эти вопросы очень важны и для новых продуктов, и для существующих, которые ты хочешь улучшить. Как узнать, нужен ли людям твой продукт и понравится ли он им? Спроси их! Покажи своим родным, друзьям или учителям набросок своего изобретения или кратко опиши его идею и посмотри, как они отреагируют. Загорятся энтузиазмом? Или пожмут плечами и что-то невразумительно промычат? Запиши все отзывы, постарайся их учесть и внеси изменения в свой замысел.

**ОТВЕТ: ДА.
КТО БУДЕТ ИМ
ПОЛЬЗОВАТЬСЯ?
ПРОДОЛЖАЙ
ИССЛЕДОВАНИЕ.**

> ДАЛЕЕ

Людам нужно твоё изобретение — это прекрасно! А теперь давай подумаем, кто эти люди. Кто и как будет пользоваться твоей программой, приложением или роботом? Если ты захочешь приспособить продукт для определенной группы людей, как это скажется на его внешнем виде и исполнении? А на его работе? Например, правила игры на правописание для дошкольников и головоломки для подростков будут заметно различаться. Подумай, на каких пользователей ты ориентируешься, и отрази это в своем проекте.

**ОТВЕТ: НЕ УВЕРЕНА.
ПРИСЛУШАЙСЯ К ОТЗЫВАМ
И ПОПРОБУЙ
ПОСЛЕДОВАТЬ ИМ.**

> НАЧАТЬ ЗАНОВО



С ЧЕГО НАЧАТЬ?

УРОВЕНЬ 4:
ХОЧУ ЛИ Я ВЗЯТЬСЯ
ЗА ЭТОТ ПРОЕКТ?



Это самый важный вопрос. Спроси себя: «Интересен ли мне этот проект настолько, чтобы осуществить его?» Хватит ли у тебя времени и сил? Это небольшое приложение, и ты успеешь его написать в перерывах между футбольными тренировками, уроками фортепиано и домашними заданиями? Или оно настолько большое, что поглотит все твое свободное время? Убедись, что у тебя найдется время, чтобы заниматься проектом. И самое важное: эта идея действительно тебя увлекает? Возможно, существует потребность в приложении-напоминалке, что пора подпилить ногти на ногах, но это не значит, что ты готова сидеть над ним ночами и по выходным. Любой проект требует затрат времени. Если ты всерьез намерена за него взяться, нужно верить в то, что ты делаешь, и иметь достаточно времени и сил, чтобы довести дело до конца.

ОТВЕТ: ДА.
ПОЗДРАВЛЯЮ!
> ПЕРЕЙТИ НА СЛЕДУЮЩИЙ
УРОВЕНЬ

ОТВЕТ: НЕ УВЕРЕНА.
ПРОДОЛЖАЙ МОЗГОВОЙ
ШТУРМ.
> ВЕРНУТЬСЯ К НАЧАЛУ



ВИЗУАЛИЗАЦИЯ УСПЕХА

Итак, полагаю, теперь у тебя есть идея. Может быть, эта идея не оригинальна, но ты задумала интересный ход и знаешь, как сделать продукт более эффективным для конкретного пользователя. Ты устроила опросы, провела исследование и готова приступить к работе.

Теперь представь, что тебе дали мозаику из 1000 кусочков, но на коробке нет примера собранной картинки. Насколько сложнее было бы собирать мозаику без картинки-подсказки? Именно поэтому следующий важный этап нового проекта — это **ВИЗУАЛИЗАЦИЯ** идеи

в форме рисунка или диаграммы. Иными словами, пришло время дизайна!

ДИЗАЙНЕРЫ. *Посмотри в Google,*

кто такой дизайнер. Ты увидишь вот что:

«Это человек, который намечает, как продукт будет выглядеть или работать, еще до его создания, обычно с помощью подробных зарисовок».

Дизайнеры могут разрабатывать костюмы, графическое оформление или внешний вид изделия. Если

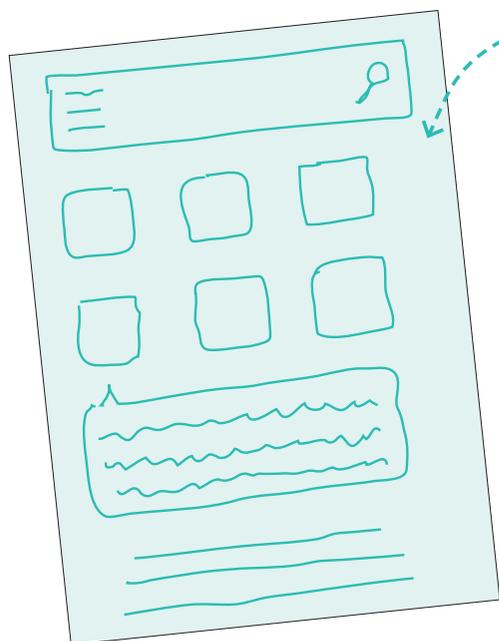
что-то можно создать, значит,

кто-то создавал для этого дизайн. В компьютерной индустрии дизайнер не обязательно является программистом. Некоторые дизайнеры специализируются на **ПОЛЬЗОВАТЕЛЬСКОМ ИНТЕРФЕЙСЕ**, то есть создают продукты, учитывая, как люди будут их использовать и взаимодействовать с ними. Есть также дизайнеры графического оформления и аппаратного обеспечения. Дизайнерам программного обеспечения не обязательно уметь программировать (хотя, если они умеют, это замечательно!), но для программистов очень важно разбираться в дизайне.

С ЧЕГО НАЧАТЬ?

МАКЕТЫ

МАКЕТ — это зарисовка будущей страницы нашего сайта или приложения. Макеты могут быть совсем простыми или очень подробными, но в любом случае они помогают нам представить, как будет выглядеть готовый продукт.



Нарисуй свой макет от руки или на компьютере. Если захочешь, можешь сделать его очень подробным!

МЕНЮ		
ФОТО + ЗАГОЛОВОК		
ФОТО + ССЫЛКА 1	ФОТО + ССЫЛКА 2	ФОТО + ССЫЛКА 3
ФОТО + ССЫЛКА 4	ФОТО + ССЫЛКА 5	ФОТО + ССЫЛКА 6

РАСКАДРОВКИ, НАБРОСКИ,
СХЕМЫ, РИСУНКИ

Раскадровка — это фактически комикс. Она показывает, как работает твоё приложение. Ее часто используют при разработке игр: раскадровка действия помогает прорисовать уровни и определить внешний вид

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

игры. Можно сделать ее совсем простой, наметив только расположение и правила уровня, или указать все детали, такие как внешность и действия персонажей.

Даже если ты делаешь не приложение или игру, а что-то другое, наглядное представление все равно очень полезно. Придумала робота? Нарисуй, как он может выглядеть. Составь схему команд, которые он будет понимать. Обозначь на схеме, как робот будет реагировать на команды. Делаешь интерактивное украшение или умные часы? Нарисуй, какой они будут формы, какого цвета и размера — а может быть, размер можно будет регулировать? Добавь к картинке примечания. Опиши свойства и функции устройства, которые будешь программировать. Что бы ты ни выбрала, наглядное представление идеи поможет тебе, когда придет время браться за ее воплощение.



ПОЛЗУЧИЙ УЛУЧШИЗМ

На стадии разработки нам осталось обсудить еще одну деталь, способную напугать даже самых опытных программистов, — это жуткое чудовище, известное как **ПОЛЗУЧИЙ УЛУЧШИЗМ**.

Ползучий улучшизм начинается, когда ты слишком увлекаешься своим

С ЧЕГО НАЧАТЬ?

проектом и постоянно добавляешь к нему что-то новое. («Щенки на прогулке», обратите внимание!) Конечно, хочется добавить и вон тот симпатичный сервис, и вот это замечательное свойство, но это все равно что положить слишком много наполнителей в мороженое: в конечном счете ты просто не сможешь его есть. Чем больше ты украшаешь свой проект, тем выше риск, что он станет слишком сложным для пользователей. А написать для него код и вовсе станет непосильной задачей. Лучше начать с простого и сосредоточиться на самых важных свойствах, необходимых для работы программы или устройства. А как только оно заработает, можно будет подумать о дополнениях.

Представь, что на уроке математики учитель просит тебя придумать пример, в ответе которого будет 4. Можно написать $1 + 1 + 1 + 2 - 1 = 4$, но можно и $1 + 3 = 4$, $2 + 2 = 4$ или $2^2 = 4$. К одному и тому же ответу можно прийти разными путями, но некоторые проще и эффективнее других.

Так же и с дизайном проекта. Ищи самый четкий и прямой путь к конечному результату. Если дизайн простой, ты сможешь уделить больше внимания тому, чтобы изобретение работало как следует и хорошо выглядело. Сложный дизайн может затянуть тебя в трясину дополнительных свойств.

Итак, поскольку разработка уже началась, самое время подумать о том, как ты будешь воплощать свою идею. Это следующая стадия работы над проектом. Здесь речь идет не о том, как будет выглядеть результат, а о том, как его осуществить и на каком языке писать программу.

Поэтому давай попрощаемся с мозговым штурмом и дизайном и скажем: «HELLO, WORLD!»



Маффины с арахисовым маслом





РАЗГАДАЙ КОД

ИТАК, С МОЗГОВЫМ ШТУРМОМ И ДИЗАЙНОМ РАЗОБРАЛИСЬ.
Пора приступать к работе?

Почти. Прежде чем начинать писать собственно программу, нужно разобраться еще с парой важных моментов. Они помогут тебе заранее увидеть возможные проблемы, исправить ошибки и в будущем сэкономят немало времени и сил.

ИДИ ПО СТРЕЛКЕ

Любая работа начинается с плана. Писатель составляет план будущего произведения, художник делает наброски, инженер пользуется чертежами, а портные следуют выкройкам. Так же и в программировании, только здесь план называется **БЛОК-СХЕМОЙ**. Блок-схема похожа на нарисованный алгоритм: на ней видны все шаги, которые нужно пройти, чтобы программа достигла своей цели. Блок-схему можно нарисовать на бумаге, на доске или привести в качестве пояснений к макету. Главное — записать ее и внимательно к ней присмотреться.

СТОП, А КАКАЯ
У ПРОГРАММЫ ЦЕЛЬ?



Цель программы — это то, что тебе от нее нужно, задача, которую она должна выполнить. Точно так же, как мозговой штурм помогает тебе понять, что ты хочешь создать, блок-схема помогает увидеть, как это должно работать. Особенно со сложными алгоритмами.

Давай попробуем составить блок-схему. Что бы нам такого сделать?

РАЗГАДАЙ КОД

СОФТБОЛЬНУЮ² ИГРУ
ДЛЯ СМАРТФОНА?

Прекрасно. Итак, представим, что мы уже сделали макет и знаем, как должна выглядеть наша игра. Теперь подумаем, что в ней будет происходить. С чего начинается софтбольный матч?



ИГРОК БЕРЕТ БИТУ!



СТОЙТЕ-КА, ЭТО НЕ ТАК
ПРОСТО. ПОСЛЕ ВЫХОДА
НА БАЗУ МНОГОЕ МОЖЕТ
СЛУЧИТЬСЯ.



Верно. Можно выбить сингл, дабл, трипл⁶ или хоум-ран⁷. Можно получить страйк-аут⁸. Питчер⁹ может подать бол¹⁰, игрок во внешнем поле может поймать

⁵ Софтбол (англ. softball — буквально «мягкий мяч») — вид бейсбола. В софтболе мяч легче, а бита и поле меньше, чем в бейсболе. — *Примеч. пер.*

⁶ Сингл, дабл, трипл (англ. single, double, triple) — удары по мячу игрока с битой (бьющего), после чего он добегаёт до первой, второй или третьей базы соответственно. — *Примеч. пер.*

⁷ Хоум-ран (англ. home run) — удар, в результате которого бьющий пробегает через все базы и возвращается на основную базу, в «дом» (англ. home). — *Примеч. пер.*

⁸ Страйк-аут (англ. strike out) — ситуация, когда бьющий выбывает из игры, получив три страйка. Самый частый случай страйка — когда бьющий не попадает по мячу. — *Примеч. пер.*

⁹ Питчер (англ. pitcher, от pitch — «бросать») — игрок команды противника, подающий мяч. — *Примеч. пер.*

¹⁰ Бол (англ. ball) — ситуация, когда питчер подает мяч вне зоны удара и бьющий не касается его битой. — *Примеч. пер.*

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК



высоко отбитый мяч, или игрок, находящийся между второй и третьей базами, может осалить бьющего прежде, чем тот доберется до базы...

Это важно: если бы мы действительно попытались воспроизвести софтбольный матч, нам потребовался бы очень сложный алгоритм, который пришлось бы писать неделями. Пожалуй, к такому мы еще не готовы, поэтому прежде, чем чертить блок-схему, попробуем немного упростить правила. (Видишь? Мы обнаружили и решили проблему еще до того, как начали писать программу. Вот для чего нужно планирование!)

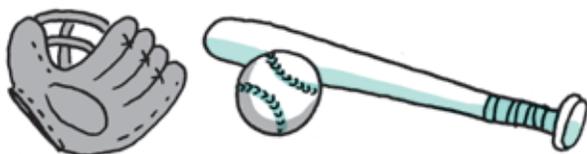
РАЗГАДАЙ КОД

Итак, УПРОЩЕННЫЕ ПРАВИЛА для игры в софтбол.

- 1 Первым битом берет игрок из команды гостей.
- 2 У каждого игрока будет три попытки, чтобы отбить мяч. Если игрок попадает по мячу, его команда получает очко. Если игрок промахивается, он получает страйк. Три страйка — игрок выбывает.
- 3 Три выбывших игрока — и бита переходит к другой команде.
- 4 Каждая команда может брать биты по семь раз (всего 14 раундов).
- 5 В конце игры смотрим, кто победил.



А теперь переведем это в блок-схему.



★НАЧАЛО★

Игра начинается! У каждого игрока есть три попытки. Задаем переменным scoreForAwayTeam, scoreForHomeTeam и turnNumber значение 0. Команда гостей начинает игру, поэтому задаем переменной currentTeamAtBat значение "гости".

НАШИ ПЕРЕМЕННЫЕ

scoreForAwayTeam
(очкиКомандыГостей)
scoreForHomeTeam
(очкиКомандыХозяев)
numberOfOuts
(числоВыбывших)
numberOfStrikes
(числоСтрайков)
currentTeamAtBat
(бьющаяКоманда)
numberOfTurns
(числоРаундов)

Раунд начинается!
Проверяем, чтобы у переменных numberOfOuts и numberOfStrikes было значение 0.

numberOfOuts меньше 3?

→ НЕТ →

Конец раунда.
Увеличиваем числоРаундов на 1.

↓ ДА ↓

Бьющая команда выставляет игрока.

numberOfStrikes меньше 3?

← НЕТ ←

Игрок выбывает!
Увеличиваем numberOfOuts на 1.

↓ ДА ↓

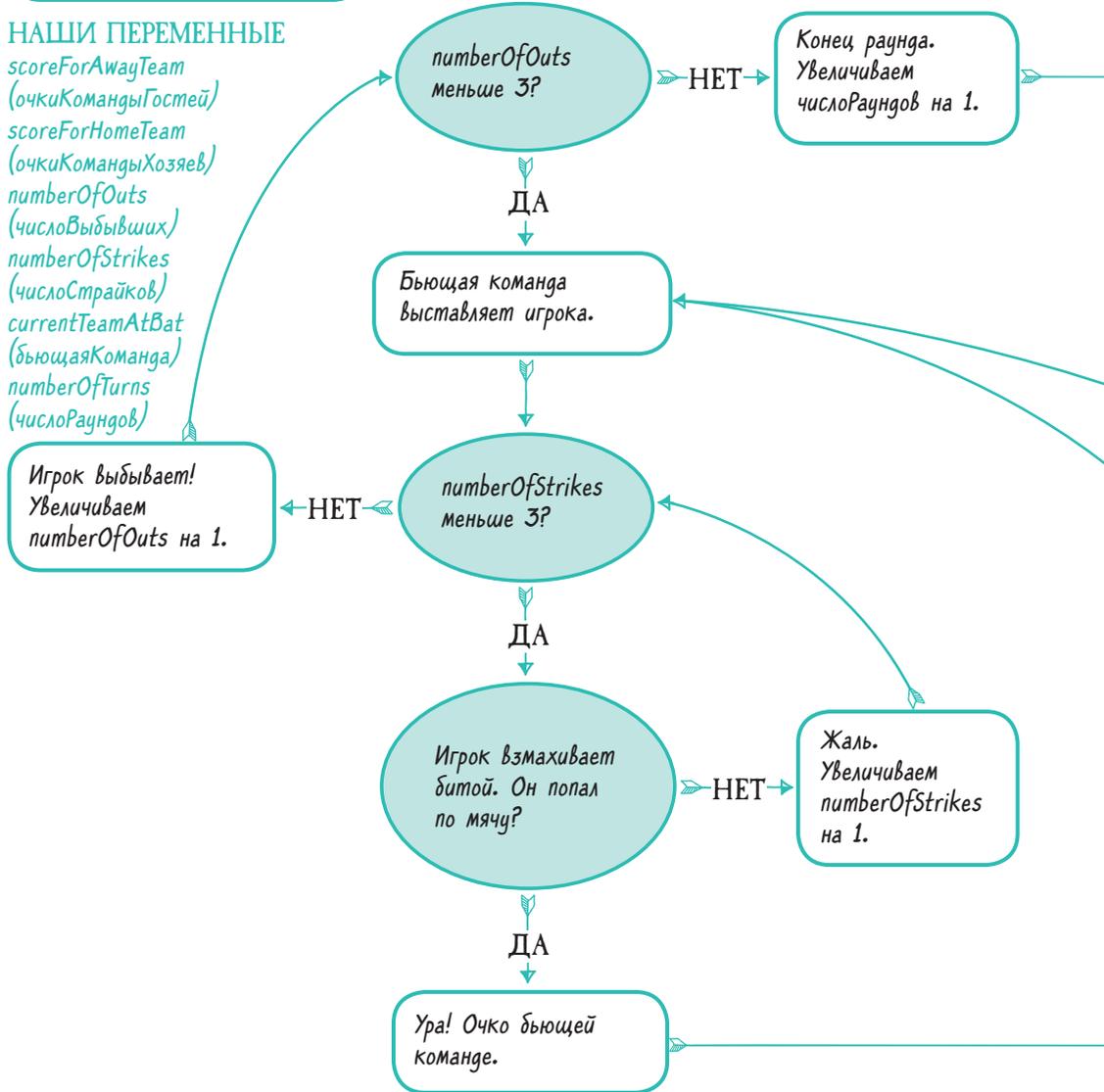
Игрок взмахивает битой. Он попал по мячу?

→ НЕТ →

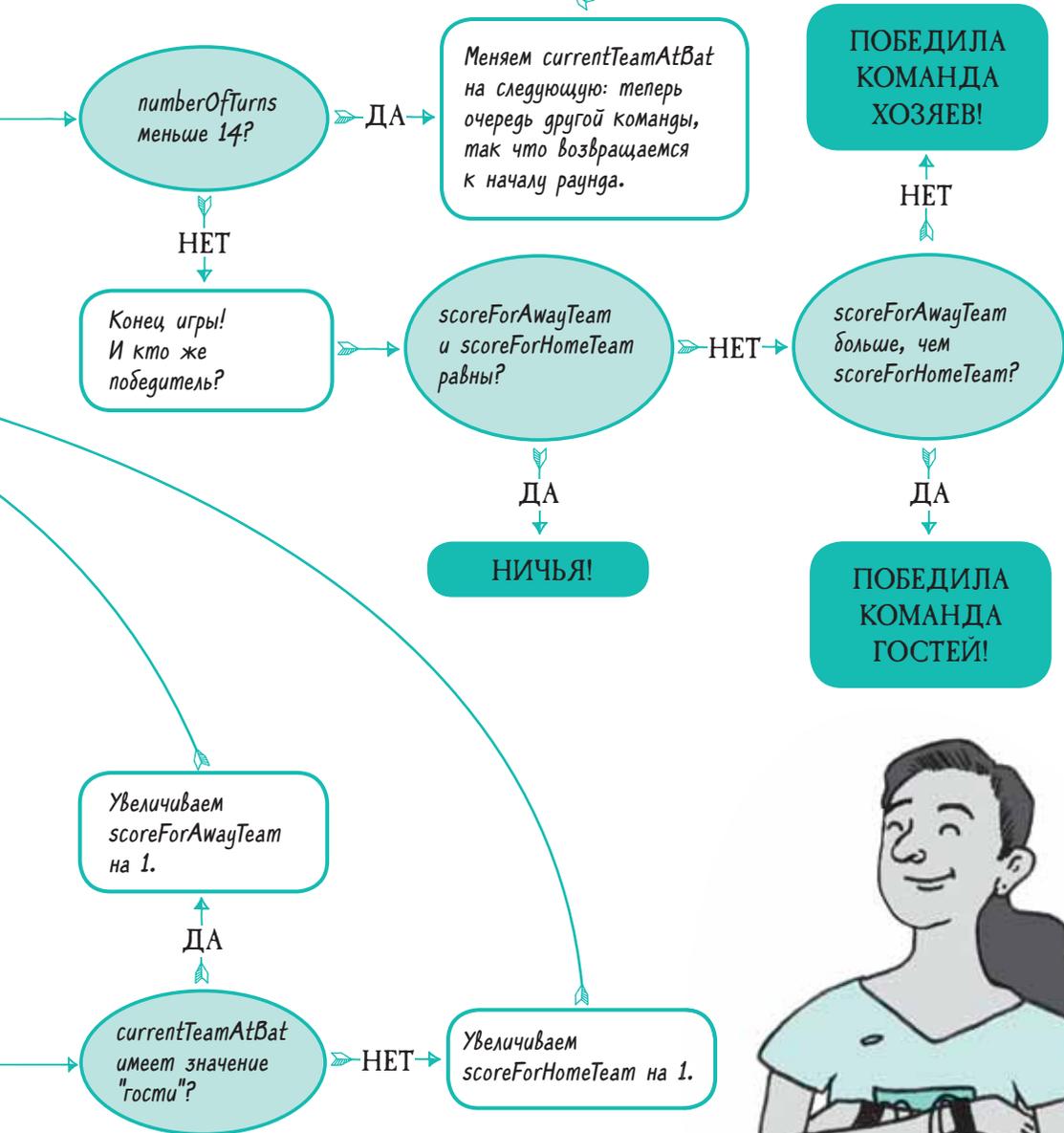
Жаль.
Увеличиваем numberOfStrikes на 1.

↓ ДА ↓

Ура! Очко бьющей команде.



СОФТБОЛ (УПРОЩЕННЫЙ)



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Что первым делом бросается в глаза?

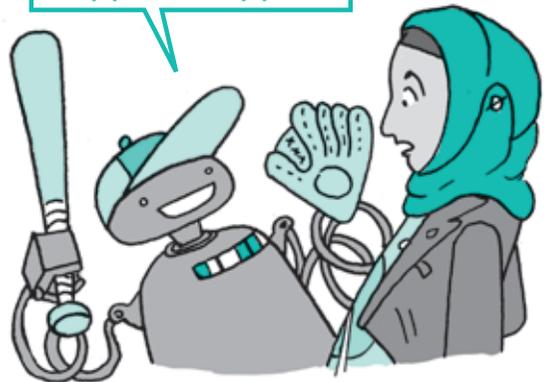


Верно. Алгоритмы редко бывают простыми и прямолинейными. Чаще всего блок-схема напоминает карту линий метро или железнодорожных путей. Все начинается с одной линии, но потом возникает множество развилок, ведущих в разные направления, в зависимости от того, что делает пользователь (попадает по мячу, выбывает из игры и так далее). Как в книгах из серии «Выбери себе приключение»: каждый выбор приводит к новой концовке. Поэтому, когда пишешь программу, недостаточно написать начало, середину и конец процесса. Твой алгоритм должен учитывать все возможные варианты. Именно для этого нам нужна блок-схема: чтобы увидеть, что еще может произойти в программе, и понять, как нам превратить это в код.

ЗАПРАШИВАЮ ДОСТУП ДЛЯ УЧАСТИЯ В МНОГОПОЛЬЗОВАТЕЛЬСКОМ МЕРОПРИЯТИИ С ПРИМЕНЕНИЕМ НЕТВЕРДЫХ ШАРОВ.

ТО ЕСТЬ... ТЫ ХОЧЕШЬ ПОИГРАТЬ В СОФТБОЛ?

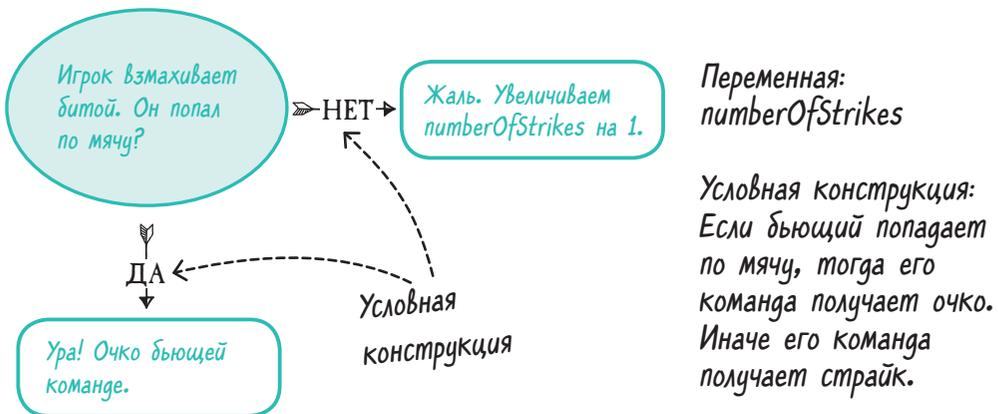
ПОДТВЕРЖДАЮ.



РАЗГАДАЙ КОД

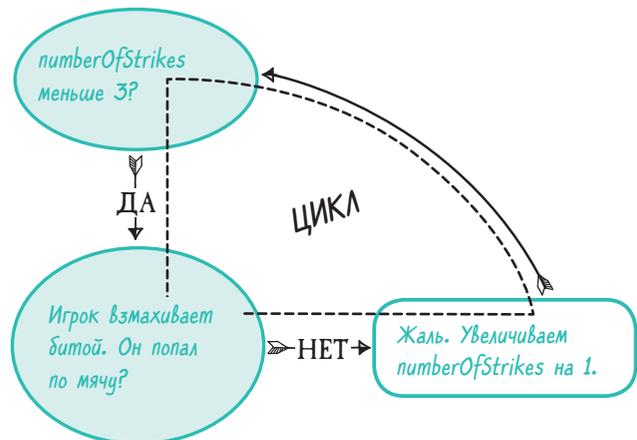
РАЗМЕЧАЕМ ДОРОГУ

Итак, мы определились, как будет проходить наша игра и какие в ней будут развилки. Теперь пора разобраться, где у нас условные конструкции и циклы, а также обозначить переменные.



Переменная: numberOfStrikes

Цикл: игрок пытается попасть по мячу до тех пор, пока не попадет или пока numberOfStrikes не станет равным 3.



ЗАПИСЫВАЕМ РЕЗУЛЬТАТ

Самый сложный этап разработки позади! Мы создали схему будущей программы и разметили ее. Самое время записать то, что у нас получилось. Это не значит, что мы уже начинаем программировать, потому что сейчас мы воспользуемся псевдокодом. Ты уже не раз встречала псевдокод в этой книге (например, обеденный алгоритм в третьей главе) и наверняка помнишь, как он работает: обычным человеческим языком мы записываем то, что позже будем программировать. Мы составляем очень подробную инструкцию, как будто пишем алгоритм на языке роботов. Тут четко видны все условные конструкции, циклы и переменные.



НО ЗАЧЕМ ТРАТИТЬ ВРЕМЯ
НА КАКОЙ-ТО ПСЕВДОКОД?
У НАС УЖЕ ЕСТЬ БЛОК-СХЕМА,
ГДЕ ВИДНО, ЧТО КУДА ВЕДЕТ, —
ПОЧЕМУ НЕ НАЧАТЬ СРАЗУ
ПРОГРАММИРОВАТЬ?

С помощью псевдокода можно расписать **ЛОГИКУ** программы, то есть все этапы программы от начала до самого конца. При этом можно не учитывать **СИНТАКСИС** языка программирования, то есть не думать, как пишется то или иное слово, где поставить пробел и как отформатировать текст.

Если ты сначала подробно запишешь блок-схему с помощью псевдокода, то во время написания кода на языке программирования тебе не придется помнить, что будет дальше или как это сработает.

РАЗГАДАЙ КОД

У тебя будет только одна задача — перевести псевдокод в настоящий код.

На старте у каждой команды 0 очков.
Начинает команда гостей.

```
scoreForHomeTeam = 0
scoreForAwayTeam = 0
```

```
turnNumber = 0
```

←----- Начинается нулевой раунд.

```
teamAtBat = "гости"
while (turns < 14):
    outs = 0
```

Пока число раундов меньше 14
(по семь на каждую команду),
команды будут меняться.

```
while (outs < 3):
```

←----- Пока из игры не выйдут
три игрока.

```
    strikes = 0
    player.next()
```

Бьющая команда будет
выставлять игроков. У них
еще нет страйков.

Пока у игрока не наберется три
страйка, он будет пытаться
попасть по мячу.

```
    while (strikes < 3):
        hit = player.bats()
        if (hit == True):
            if (currentTeamAtBat == "гости"):
                scoreForAwayTeam = scoreForAwayTeam + 1
            else:
                scoreForHomeTeam = scoreForHomeTeam + 1
        else:
            strikes = strikes + 1
```

Если игрок попадает по мячу,
его команда получает очко.

Если игрок промахивается,
он получает страйк.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК



СЕКРЕТНЫЙ ИНГРЕДИЕНТ: РЕШЕНИЕ ЗАДАЧ

Когда ты готовишь торт по рецепту, шьешь по выкройке, раскрашиваешь набросок или собираешь игрушку, ты делаешь то, что очень похоже на программирование. Каждое из этих занятий — поэтапный процесс и выполняются по определенному плану.



М-М . . .
Я НЕ ОЧЕНЬ ПОНИМАЮ, ЧТО
ТАКОЕ ЯЗЫК ПРОГРАММИРОВА-
НИЯ И ПОЧЕМУ ИХ ТАК МНОГО.

РАЗГАДАЙ КОД

Помни, что по сути программирование — это решение задач. Ты берешь сложную задачу и, применяя вычислительное мышление, разбиваешь ее на меньшие задачки, с которыми справиться проще. Останется только выбрать язык программирования, и можно приступать.

Поэтому давай выберем язык.

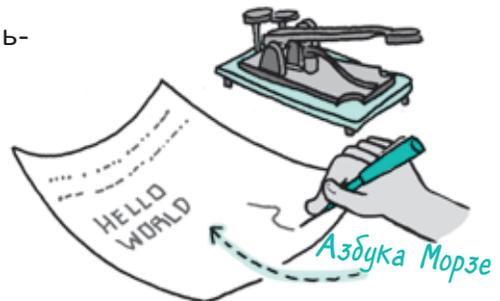
На уровне механизма большинство компьютеров работают на очень простом языке. Он называется **ДВОИЧНЫМ КОДОМ**. Двоичный код превращает числа, слова и даже звуки и картинки в серии единиц и нулей, которые поступают в компьютер в виде электрических импульсов. Каждый разряд такого **ДВОИЧНОГО ЧИСЛА** называется битом.

А ПОЧЕМУ КОМПЬЮТЕРЫ
ИСПОЛЬЗУЮТ **ДВОИЧНЫЕ**
ЧИСЛА, А НЕ ОБЫЧНЫЕ ЧИСЛА
И СЛОВА, КАК МЫ?



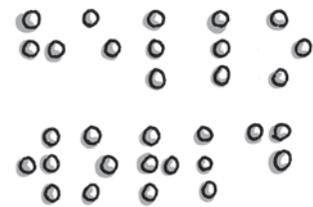
Компьютеры собраны из миллионов электрических цепей. У этих цепей, как у выключателей, есть только два положения: «включено» и «выключено». С помощью всего двух цифр — 1, то есть «включено», и 0, то есть «выключено», — компьютеры способны свести миллиарды различных комбинаций чисел и команд к последовательностям всего из двух электрических импульсов.

Что такое код — сам по себе, не только программный? В широком смысле код — это использование слов, чисел, букв и других символов для того, чтобы обозначить что-то, не имеющее к этим символам прямого отношения.



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Часто за кодом скрывается некая тайна, истинный смысл слов: так военные и шпионы кодируют свои сообщения. Но код может и упростить слишком сложную фразу или информацию, чтобы общение шло быстрее. Хороший пример — морские сигнальные флаги: за счет разных цветов и форм они способны легко доносить смысл целых предложений на больших расстояниях в море.



Шрифт
Брайля

То же делает и двоичный код — он упрощает сложную систему сообщения всего до двух символов. Существует великое множество двоичных кодов, которые используются за пределами компьютерной индустрии. В азбуке Морзе, которую применяли в первых телеграфных линиях и на кораблях, слова и фразы можно записывать точками и тире.

Текст, записанный азбукой Морзе, можно передавать с помощью кратких и долгих звуков или вспышек света. Шрифт Брайля — алфавит, который слепые и слабовидящие люди читают на ощупь. В шрифте Брайля все символы кодируются как серии выпуклых точек и пропусков.



ТО ЕСТЬ ВСЕ ЭТИ
СЛОЖНЕЙШИЕ, ЗАПУТАННЕЙШИЕ
КОМПЬЮТЕРНЫЕ ОПЕРАЦИИ —
ВСЕГО-НАВСЕГО ВКЛЮЧЕНИЕ
ИЛИ ВЫКЛЮЧЕНИЕ МИЛЛИОНОВ
КРОШЕЧНЫХ ЭЛЕКТРИЧЕСКИХ
ПЕРЕКЛЮЧАТЕЛЕЙ?

В общем, да. Поразительно, правда?

РАЗГАДАЙ КОД

ВЫ ГОВОРИТЕ ПО-ДВОИЧНОМУ?

Представь, что тебе вручную пришлось бы переводить все команды в нули и единицы. Было бы почти невозможно не запутаться. Мы привыкли говорить и писать на человеческих языках, и нам сложно читать и писать на двоичном коде. Поэтому языки программирования так важны.

Вот почему открытие, совершенное в 1952 году, навсегда изменило историю компьютеров. Именно тогда один ученый создал первый работающий **КОМПИАТОР** — специальную программу, которая переводила алфавитно-цифровые символы («а, б, в...» и «1, 2, 3...») на двоичный язык компьютеров. С помощью компилятора компьютеры смогли понимать язык людей. Это была настоящая революция в компьютерном мире. Теперь инженеры могли изобретать языки программирования — «код», писать на котором и работать с которым было гораздо проще. Так кто же был этот замечательный ученый?



Первооткрывательница в области компьютеров **ГРЕЙС МЮРРЕЙ ХОППЕР**

Современного программирования не существовало бы, если бы не блестящий ум Грейс Хоппер, которую часто называют «Удивительной Грейс». В 1941 году она была замужем и преподавала математику в колледже Вассар.

Седьмого декабря 1941 года японцы напали на Перл-Харбор, и это стало решающим моментом не только для США, вынужденных вступить во Вторую мировую войну, но и для Грейс. Вскоре она развелась с мужем и поступила на флот, где применила свой незаурядный аналитический ум в военном деле. Здесь началась ее карьера специалиста по компьютерам, которая продолжилась и после окончания войны. Грейс перешла в частный сектор, изобрела компилятор и вместе с другими учеными разрабатывала языки программирования, в том числе язык Кобол. Без ее трудов не было бы возможно ни современное программирование, ни программное обеспечение, которое позволяет людям без специального образования работать на компьютере.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Благодаря Грейс Хоппер у нас есть компиляторы. Они переводят языки программирования, понятные людям, в двоичный код, доступный компьютерам.

ВЫБИРАЕМ ЯЗЫК ПРОГРАММИРОВАНИЯ

Языки программирования похожи на человеческие: их много, и каждый по-своему выражает одну и ту же мысль. Чтобы примерно представить, сколько в мире разных языков программирования, погугли «программа hello world».



РАЗГАДАЙ КОД

```
using System;
class Program
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Hello, world!");
    }
}
```

C#



Hello, World!

Java

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World");
    }
}
```

Hello, World!

Видишь, как по-разному можно написать одну и ту же фразу?

Эта простенькая программа просит компьютер написать слова «Hello World» (английское неформальное приветствие, близкое к нашему «всем привет!»). Ее используют, чтобы убедиться, что язык программирования работает нормально, и чтобы продемонстрировать, как много в мире языков.

А ЕСЛИ ЯЗЫКОВ ТАК МНОГО, КАК ПОНЯТЬ, КАКОЙ НАМ НУЖЕН?



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Зависит от того, что ты хочешь сделать: для одних задач лучше подходят одни языки, для других — другие. Четких правил, какой язык для чего использовать, не существует, но вот полезное руководство.

*Язык программирования**Где применяется***SCRATCH (СКРЕТЧ)**

Система визуального программирования. Использует цветные блоки кода и готовые графические изображения для создания простых игр и приложений. Хорошо подходит новичкам.

Компьютерные
игры, рисование,
анимация

LEGO MINDSTORMS NXT

Язык для создания программируемых роботов.

Робототехника

PYTHON

Универсальный язык программирования, подходит для разных задач, включая приложения для научных исследований.

Сайты и приложения,
разработка
программного
обеспечения, игры,
робототехника

HTML/CSS

Это языки не программирования, а разметки. Языки разметки определяют, как будет строиться и отображаться текст — например, заголовок или статья на сайте. Также с их помощью можно структурировать текст, добавлять в него что-то новое (HTML) или красиво оформлять его (CSS). Они не считаются языками программирования, так как не умеют запоминать, повторять и принимать решения, зато с их помощью можно спланировать внешний вид страницы и расположить ее элементы в нужном порядке.

Сайты
и приложения

JAVASCRIPT

Не то же самое, что Java, хотя названия и похожи. На этом языке пишут интерактивные элементы веб-страниц. Для JavaScript существует множество библиотек (самая большая из них — jQuery), которыми можно бесплатно пользоваться и добавлять туда новые элементы.

Сайты
и приложения

РАЗГАДАЙ КОД

*Язык программирования**Где применяется***ARDUINO PROGRAMMING LANGUAGE**

Создан на основе C++ (универсального языка программирования), используется в микроконтроллерах Arduino.

Робототехника

JAVA

Еще один универсальный язык, один из самых популярных в мире. На нем же пишут приложения для Android.

Сайты и приложения, игры

SWIFT

Язык приложений для iOS (наряду с XCode — средой разработки программного обеспечения для компьютеров Apple). Очень молодой язык с большими перспективами; появился только в 2014 году.

Приложения

PROCESSING

Создатели описывают его как удобный альбом для программных эскизов; прекрасно подходит для рисования и дизайна.

Компьютерные игры, рисование, анимация

C (ПРОИЗНОСИТСЯ «СИ»)

Старый и широко известный язык; хорош в ситуациях, когда важна скорость. Это язык системного программирования, то есть в нем очень мало преград между тобой и системными нулями и единицами.

Библиотеки, программное обеспечение

C# (ПРОИЗНОСИТСЯ «СИ ШАРП»)

Изначально задумывался как похожий на C, но во многом от него отличается. Применяется для разработки игр.

Компьютерные игры

MAYA EMBEDDED LANGUAGE (ВСТРОЕННЫЙ ЯЗЫК MAYA)

Используется в программе Maya. С ее помощью Pixar делает свои трехмерные анимационные мультфильмы. Позволяет писать внутри Maya скрипты на Python, чтобы создавать 3D-модели и анимацию.

Анимация, рисование, компьютерные игры

СБОРКА

С языком определились? Пора приниматься за сборку. Проще говоря, пришло время писать программу!



ПОДСКАЗКА:

Держи файлы в порядке и не забывай про резервную копию!

С началом сборки нужно будет не только создавать новые файлы, но и следить за старыми. Лучший способ облегчить себе задачу — хранить файлы в строгом порядке. Придумай для них систему названий, где указывались бы дата создания или номер версии, и следуй этой системе. Называй файлы единообразно. Каждый день делай резервную копию проекта на внешнем носителе или в онлайн-хранилище: пусть у тебя всегда будет актуальная копия вне рабочего компьютера. Так ты сможешь быстро найти то, что нужно, или исправить ошибку, как только она возникнет.

РАЗГАДАЙ КОД

И вот оно — ты программируешь! Если ты уже прошла все этапы разработки и планирования, остается только перевести алгоритмы в код. Легче легкого, верно?

Что может пойти не так?







ОТЛАДКА

ИТАК, ПРЕДСТАВИМ, ЧТО ТЫ ИЗУЧИЛА НЕСКОЛЬКО РУКОВОДСТВ в Интернете и освоила язык программирования. Ты сидишь за своим ноутбуком, с головой ушла в работу: пишешь строки, переменные и циклы, добавляешь параметры и функции. Ты программируешь как одержимая. Все получается — ты просто суперзвезда программирования! Ты дописываешь последнюю строчку и запускаешь программу...

...и, как прыщ в тот день, когда вы с классом должны фотографироваться, выскакивает оно:

ЗЛОВЕЩЕЕ СООБЩЕНИЕ ОБ ОШИБКЕ.

Значит, пришла пора заняться по-настоящему серьезным делом: в программировании это тестирование и...

ОТЛАДКА

Написать программу — это одно; совсем другое — сделать так, чтобы она работала. Когда ты впервые запустишь программу, чтобы протестировать ее, то почти гарантированно наткнешься на какую-нибудь проблему. В компьютерной среде, когда в программе что-то не работает, это называется багом (от англ. bug — «жук»). Чтобы найти, определить и исправить баг, нужно выполнить **ОТЛАДКУ**.



А ПРИ ЧЕМ ТУТ ЖУКИ?

Помнишь Грейс Мюррей Хоппер, изобретательницу компилятора? Легенда гласит, что однажды ее компьютер отказался работать из-за самого настоящего мотылька, забравшегося в механизм. Когда насекомое извлекли, все заработало. Поэтому ошибки в программе и называют жуками.

СИНТАКСИС ИЛИ ЛОГИКА?

Если в твой ноутбук никто не залетел, то, скорее всего, ошибка относится к одному из двух наиболее распространенных типов: синтаксическому или логическому.

СИНТАКСИЧЕСКАЯ ОШИБКА связана с написанием кода: ты забыла поставить какой-нибудь символ, букву, пробел или знак препинания.

ЛОГИЧЕСКАЯ ОШИБКА связана с работой программы: ты просишь компьютер сделать что-то бессмысленное или то, чего он не может.

Вот несколько примеров, написанных на языке JavaScript.

ОТЛАДКА

НЕПРАВИЛЬНЫЙ КОД

```
var myName = Гуля";
```

ПРАВИЛЬНЫЙ КОД

```
var myName = "Гуля";
```

СИНТАКСИЧЕСКАЯ ОШИБКА

В большинстве языков программирования строка — в данном случае имя Гули — должна быть заключена в кавычки с обеих сторон. Синтаксические ошибки встречаются очень часто: это может быть пропущенная точка с запятой, лишний пробел или опечатка.

НЕПРАВИЛЬНЫЙ КОД

```
var numberOfPeople = 0;
var pizzaSlices = 12;
pizzaSlicesPerPerson = pizzaSlices /
numberOfPeople;
```

ПРАВИЛЬНЫЙ КОД

```
var numberOfPeople = 4;
var pizzaSlices = 12;
pizzaSlicesPerPerson = pizzaSlices /
numberOfPeople;
```

ОШИБКА ДЕЛЕНИЯ НА НОЛЬ

У нас есть переменные для количества людей, пришедших на вечеринку, и для числа кусков пиццы. Мы делим число кусков пиццы на количество людей, чтобы определить, сколько кусков достанется каждому. Но компьютеры, как и мы, не умеют делить на ноль, поэтому неплохо бы пригласить кого-нибудь на нашу вечеринку!

НЕПРАВИЛЬНЫЙ КОД

```
if (myAge = 12) {
    alert("В следующем году я получу
    паспорт!");
}
```

ПРАВИЛЬНЫЙ КОД

```
if (myAge == 12) {
    alert("В следующем году я получу
    паспорт!");
}
```

НЕ ТОТ ЗНАК

В условной конструкции, чтобы проверить, равны ли обе части условия, нужно использовать двойной знак «равно», потому что простой знак «равно» задает переменную. В примере слева вместо проверки, является ли myAge (мойВозраст) равным 12, мы задаем myAge значение 12!

НЕПРАВИЛЬНЫЙ КОД

```
pizzaSlicesEaten = 0;
do {
    self.eat("pizzaSlice");
} while (pizzaSlicesEaten < 3);
```

ПРАВИЛЬНЫЙ КОД

```
pizzaSlicesEaten = 0;
do {
    self.eat("pizzaSlice");
    pizzaSlicesEaten = pizzaSlicesEaten + 1;
} while (pizzaSlicesEaten < 3);
```

БЕСКОНЕЧНЫЙ ЦИКЛ

В этом цикле мы едим пиццу до тех пор, пока pizzaSlicesEaten (съеденоКусковПиццы) меньше 3. Но в варианте слева мы застреваем в бесконечном цикле, потому что забыли, что нужно увеличивать pizzaSlicesEaten с каждым прохождением цикла. Эту пиццу мы будем есть вечно!

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК



И КАК
ИХ ИСПРАВИТЬ?

Есть несколько полезных способов. Но прежде всего баг нужно найти. Хочешь проверить, не закралась ли в твой код ошибка? Запусти программу. Если что-то не так, то, как правило, язык программирования, на котором ты пишешь, выдаст сообщение об ошибке: в нем будет сказано, где и почему эта ошибка происходит.

Потом следует выяснить, какую ошибку ты допустила: синтаксическую или логическую. Если программа работает, но не делает того, что должна, это логическая ошибка. Если программа не работает, потому что в коде опечатка, это синтаксическая ошибка. Сообщение об ошибке может здорово помочь, но нужно еще понять, что в нем сказано.

А КАК ЭТО
ВЫЯСНИТЬ?

Помни, что ты в любой момент можешь попросить о помощи. Попробуй воспользоваться нашим руководством о том, когда и как это лучше делать.



ОТЛАДКА

Кого и когда просить о помощи

ВО-ПЕРВЫХ: ТВОЙ ГЕНИАЛЬНЫЙ УМ

Именно. Сначала спроси себя. Посмотри, удастся ли тебе самой справиться с проблемой. Некоторые сообщения об ошибках не требуют пояснений. Кроме того, иногда можно обнаружить ошибку, внимательно изучив строчку кода, на которую указывает сообщение.

ПОТОМ: ИНТЕРНЕТ

О Интернет, дивный мир поисковых систем, форумов, технических блогов и видеопроцедур! Там уж точно найдется решение. Погугли сообщение об ошибке — может быть, тебе подскажут, как с ней справиться. Или можно заглянуть в ДОКУМЕНТАЦИЮ к языку программирования, то есть в его инструкцию по эксплуатации, где указаны разные технические подробности. Ссылка на документацию обычно бывает на официальном сайте языка.



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

ЗАТЕМ: ДРУЗЬЯ

Проблема так и не решилась? Поговори с другом или подругой, особенно если вы работали вместе или ваши проекты чем-то похожи. Возможно, ему (ей) уже попадалась похожая ошибка и удалось найти решение. Думаю, тебе с радостью помогут, ведь для этого и нужны друзья?

И НАКОНЕЦ: УЧИТЕЛЬ ИЛИ СТАРШИЙ ТОВАРИЩ

В конце концов, ты всегда можешь попросить помощи у учителя или кого-нибудь из взрослых, кто разбирается в программировании: старшего брата или сестры, друга, соседа или знакомой из «Программирования для девочек». Учителя, особенно преподаватели программирования, не только знают много полезных вещей, но и с радостью помогают ученикам преодолевать трудности и учиться чему-то новому (попробуй сама и убедись!)

НО ПОЧЕМУ ИМЕННО
В ТАКОМ ПОРЯДКЕ? ЕСЛИ
УЧИТЕЛЯ ВСЕ ЗНАЮТ,
ПОЧЕМУ НЕ ПОЙТИ СРАЗУ
К НИМ?



Давай представим, что через 20 лет ты нашла работу своей мечты (может, даже в компьютерной сфере) и получаешь новое интересное задание: помочь шефу с крупным и важным проектом. Но в ходе работы ты сталкиваешься с чем-то, что тебе непонятно. Ты будешь писать шефу каждый раз, когда у тебя возникает вопрос? Или попробуешь разобраться сама, поискав информацию по этому вопросу или спросив у коллег?

ОТЛАДКА



Если ты чего-то не понимаешь, всегда лучше обратиться за помощью, чем делать вид, что у тебя все в порядке. Но не менее важно научиться решать проблемы самостоятельно. Справляясь с проблемами лично, ты начинаешь лучше разбираться в том, что делаешь, учишься брать на себя ответственность и становишься более уверенной в себе всякий раз, когда удастся решить хитрую задачку без посторонней помощи.

Итак, как нам найти ошибку? Есть несколько способов.

РЕЗИНОВАЯ УТКА СПЕШИТ НА ПОМОЩЬ!

Возникла проблема? Попробуй «метод резиновой утки»: расскажи об этой проблеме любому, кто готов выслушать. Обратись к другу, маме, маме друга, почтальону. В крайнем случае посади на стол резиновую утку и поговори с ней. Может, тебе и не посоветуют ничего полезного — дело не в ответе. Иногда, проговорив



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

проблему вслух, ты замечаешь возможность решить ее, на которую раньше не обращала внимания. Точно так же, как художник отходит от мольберта, чтобы окинуть взглядом картину и сделать очередной мазок, рассказ о проблеме помогает отвлечься от деталей, увидеть ситуацию целиком и, возможно, найти ответ.

ДЕТЕКТИВ В ПОИСКАХ ОШИБКИ



Здесь тебе понадобится свой указательный палец, бумага и ручка. Мы займемся старой доброй детективной работой. Проверь свой код с самого начала и води по строчкам пальцем. Ищи синтаксические ошибки. Составь список имен переменных в коде и смотри, что с этими переменными происходит в программе: следи за тем, чтобы они оставались верными и не менялись ни с того ни с сего. Сравни код с макетом и блок-схемой, посмотри, нет ли ошибок в логике и не упустила ли ты мелкую деталь.

ФУНКЦИЯ ВЫВОДА

Обычно в языках программирования есть функция, которая называется «вывод» (англ. `print`) или примерно так; точное название можно найти в документации или просто в Интернете. С ее помощью можно в любой момент проверять значения интересующих тебя переменных. Используй ее каждые 5–10 строчек, чтобы убедиться, что в этой части кода все верно. Можно также вывести сообщение, которое подскажет, какие условные

ОТЛАДКА

конструкции истинны. Так с помощью функции вывода можно установить, где в программе ошибка: тестируя код небольшими порциями, ты выяснешь, где что-то пошло не так.

ИНТЕГРИРОВАННАЯ СРЕДА РАЗРАБОТКИ (ИСР)

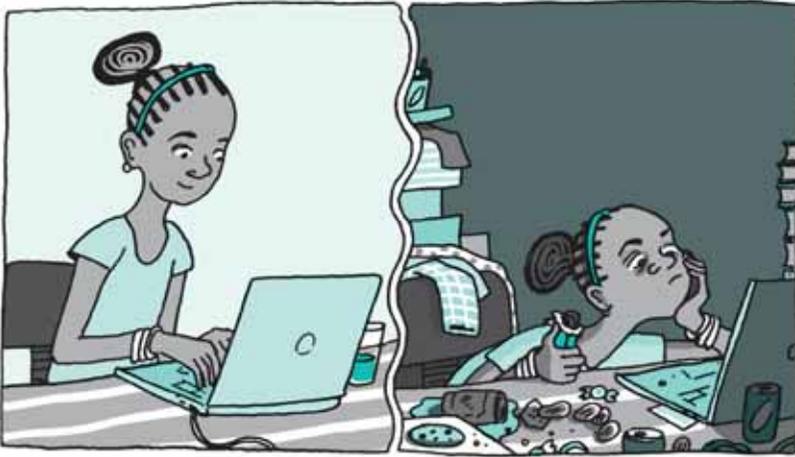
Фактически ИСР — это программа, которая помогает писать код. В нее входят **ТЕКСТОВЫЙ РЕДАКТОР** (окно, где можно писать), компилятор и собственная система отладки кода. Это очень полезная вещь. Большинство ИСР автоматически распознают синтаксические ошибки и выделяют их. Так ты сразу видишь, что сделала опечатку, а не ищешь ее потом по всей программе. Также в них есть автодополнение ввода и другие автоматизирующие функции, помогающие писать быстрее и делать меньше ошибок и опечаток.



СДЕЛАЙ ПЕРЕРЫВ

Если уж совсем ничего не получается, бывает полезно сходить прогуляться, перекусить или даже вздремнуть. Пусть внешне мы ничем не заняты — это вовсе не значит, что наш мозг перестал просчитывать варианты и искать ответы. На самом деле полноценный отдых, здоровое питание, общение с родными и близкими, свежий воздух и физические упражнения — ключ к тому, чтобы справиться с любыми трудностями. Все это необходимо для того, чтобы мыслить ясно. На свежую голову куда легче заметить ошибку и избежать ее.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

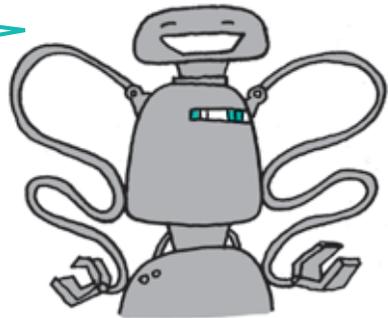


И последний — по счету, но не по важности — способ решить любую проблему — это...

ПРИМИ, ЧТО ТЫ НЕСОВЕРШЕННА

Нам, женщинам, от общества постоянно поступают противоречивые сигналы о том, что мы должны и чего не можем делать. Мы должны быть хорошими, милыми девочками, должны следовать правилам, быть дружелюбными, добрыми, готовыми прийти на помощь — словом, мы должны быть **СОВЕРШЕННЫМИ!** Но послушайте, что скажет вам взрослая, состоявшаяся женщина и мать: **СОВЕРШЕННЫХ ЛЮДЕЙ НЕТ.** На всякий случай я повторю, хотя нет, пусть лучше он скажет:

ПОДТВЕРЖДАЮ.
СОВЕРШЕННЫХ ЛЮДЕЙ
НЕТ. И РОБОТОВ ТОЖЕ.



Какое это имеет отношение к программированию? Самое прямое! Ты,

ОТЛАДКА

наверное, привыкла, что тебя хвалят, когда ты справляешься с заданием, все успеваешь и ведешь себя как «хорошая девочка». А что будет, если ты сделаешь что-то не так или ошибешься, как ошибаются все, кто чему-то учится (а иногда и без причины)?

Женщины (да и мужчины) порой бывают слишком строги к себе. Мы злимся на себя, разочаровываемся в себе, а иногда и обзываем себя или думаем что-нибудь вроде: *«Вот глупость! Как я сразу не догадалась? Что со мной не так?»* Ты бы никогда не сказала ничего подобного своей лучшей подруге, учителю или кому-то из родителей.

Тогда зачем так обращаться с собой? Если ты хочешь добиться успеха в чем-то — будь то спорт, игра на музыкальном инструменте, изучение языка, написание теста или создание робота, — не пытайся быть совершенной. Лучше будь храброй. Не бойся! Каждый раз, когда ты ошибаешься, ты учишься.

Но прежде всего *будь добра к себе самой*. Прощай себе ошибки, говори себе: *я смогу, я пойму*. Быть несовершенной не страшно. Успех приносят постоянные *тренировки, терпение и настойчивость*.

Ты сможешь. Я уверена, что где-то глубоко внутри ты уже это знаешь. Так прислушайся к этому голосу.

Помни: быть девочкой, которая умеет программировать, — значит жить этим настоящим моментом, учиться, погружаться в работу с головой. Наградой за то, что ты чему-то научилась, может быть не только конечный результат, но и сам процесс, ощущение, что ты используешь свой гениальный ум, свое мужество и свой дар ясно мыслить, чтобы разобраться в этом.

Итак, ты готова воплотить в жизнь свое собственное удивительное изобретение с помощью программирования? Давай начнем с того, что все любят, во что играют в Интернете, на телефонах, компьютерах и приставках... представляешь? Теперь ты можешь делать их сама!





КОМПЬЮТЕРНЫЕ ИГРЫ

НАВЕРНОЕ, СЕЙЧАС ТЕБЯ ПРОСТО ПЕРЕПОЛНЯЮТ ИДЕИ ДЛЯ НОВЫХ проектов, поэтому давай обсудим, какие варианты у нас есть. Что бы нам такого сделать?

я хочу узнать,
как делают
компьютерные
игры!



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

В мире столько разных игр, и чего мы в них только ни делаем: ловим зверьков, комбинируем конфеты, носимся наперегонки на машинах, возводим миры из кирпичиков, разгадываем головоломки, ищем сокровища и зарабатываем очки, чтобы перейти на новый уровень. Но в любом случае компьютерные игры — одно из самых популярных применений для программирования, и они отлично подходят для того, чтобы рассказывать истории и обучать игроков. Это хорошая идея для первого проекта: можно начать с простого концепта, а по мере того, как ты будешь обучаться программировать, добавлять новые, более сложные детали.



Это точно! Игровые **ЖАНРЫ** — это группы, к которым относятся игры в зависимости от того, как в них играют. Вот несколько жанров для примера.

КВЕСТЫ и РОЛЕВЫЕ ИГРЫ



ГОЛОВОЛОМКИ
БОЕВИКИ
СИМУЛЯТОРЫ
ГОНКИ
СТРАТЕГИИ



КОМПЬЮТЕРНЫЕ ИГРЫ

Игры бывают не только разных жанров, но и могут служить различным целям. Одни просто развлекают. Другие учат чему-то или помогают натренировать навык, как, например, детские игры для изучения алфавита, математические игры, головоломки или задачи. Третьи привлекают внимание к проблеме или привязаны к фильму, персонажу или торговой марке.

Игра может вырасти из чего угодно, чем ты интересуешься помимо **КОМПЬЮТЕРНОГО ПРОГРАММИРОВАНИЯ**. Ты хорошо пишешь? Квесты и ролевые игры требуют продуманных историй с персонажами, сценарием и диалогами. Обожаешь sudoku и кроссворды или цвета и формы? Для головоломок нужны и задачи, и дизайн, и умение программировать. Интересуешься картами, пейзажами и различными природными условиями? Все это можно найти в симуляторах.

Я И НЕ ДУМАЛА, ЧТО С ИГРАМИ
СТОЛЬКО ВСЕГО СВЯЗАНО.
ТЕПЕРЬ ЕЩЕ СЛОЖНЕЕ ВЫБРАТЬ,
С ЧЕГО НАЧАТЬ!

Давай поговорим с группой наших программисток об игре, которую они создали. Может быть, мы сможем узнать у них что-то полезное о **РАЗРАБОТКЕ КОМПЬЮТЕРНЫХ ИГР**.

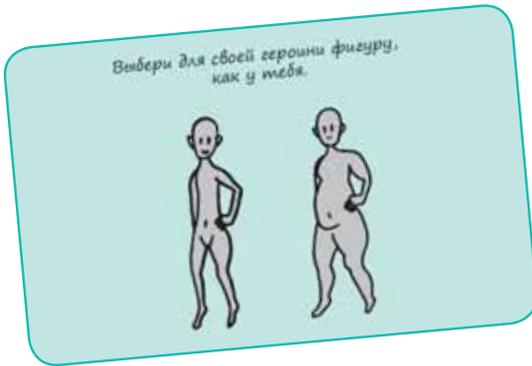


ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Карьера от кутюр

Эта игра стала выпускной работой Глори К., Захры Л., Марии М. и Нани Н., ходивших на курсы «Программирования для девочек» в Нью-Йорке. «Карьера от кутюр» — это игра-одевалка для девочек от 5 до 12 лет. Игра показывает, что можно быть красивой *независимо от того*,

какие у тебя фигура, прическа или цвет кожи. Здесь можно настроить цвет кожи, фигуру, прическу и профессию и узнать интересные факты о женщинах, которые работали в выбранной тобой области.



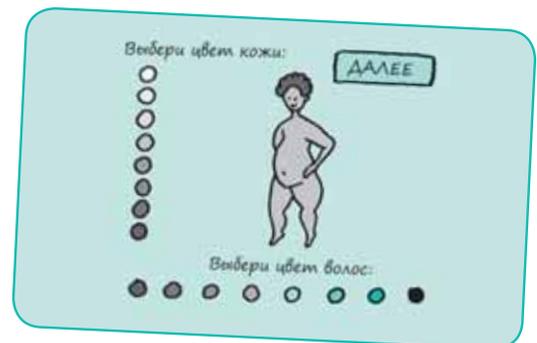
Расскажите, как вы придумали эту игру.

ГЛОРИ: Мы оказались в одной команде. Я хотела сделать что-нибудь такое, что придало бы девочкам уверенности в себе. Мария, Захра и Нани поддержали меня. По-моему, это Мария придумала игру-одевалку.

Почему именно игра-одевалка?

МАРИЯ: Когда я была маленькой, я обожала играть в одевалки в Интернете, но герои там не были похожи на реальных людей. Поэтому мы решили, что в нашей игре будут все возможные расы и стили.

ЗАХРА: Я темнокожая, и в детстве я видела, что в большинстве одевалок, даже если и была темнокожая девочка, оттенок кожи был всего один — ни темнее, ни светлее, никаких вариаций. А волосы всегда были прямыми, не как у меня. У меня вьющиеся волосы, но в играх я таких



КОМПЬЮТЕРНЫЕ ИГРЫ

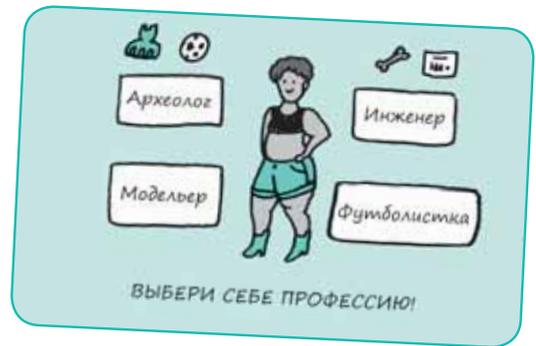
не видела и не понимала, что это красиво. Поэтому я решила добавить в нашу игру природные типы волос, чтобы показать девочкам: не обязательно иметь прямые волосы, чтобы быть красивой.

Расскажите, почему вы решили добавить в игру выбор профессии.

МАРИЯ: Мы хотели сделать игру не только про внешность, но и про устремления: чего ты можешь достичь и из чего можешь выбирать.

ГЛОРИ: Мы хотели, чтобы наша игра была не похожа на другие одевалки. Есть, конечно, одевалки с поп-звездами или принцессами, но мы хотели, чтобы вариантов было много.

Мы хотели показать, что не бывает неправильного выбора. Что бы ты ни выбрала, ты можешь добиться успеха!



Расскажите, как вы создавали свою игру: какой язык выбрали, с чего начинали?

ЗАХРА: Мы начали с псевдокода, записали его на доске. Тогда мы еще ничего не программировали — просто обсуждали, решали, какую программу нам взять.

МАРИЯ: Мы использовали JavaScript и немного HTML, потому что решили сделать сайт.

ГЛОРИ: Нани умеет рисовать, так что все иллюстрации — ее.

То есть вы писали не только программный код, но и сами создавали рисунки? Вы все делали вместе?

МАРИЯ: Рисовать умела только Нани, поэтому мы рассказали ей, чего хотим. Она все соединила, добавила свой собственный стиль, и получилось очень здорово.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

ГЛОРИ: А программу мы и правда писали все вместе. Слушали музыку, болтали и писали.

ЗАХРА: Было очень весело.

МАРИЯ: По-моему, мы чуть ли не каждую строчку вместе писали. Мы сидим за соседними компьютерами, я говорю: «Слушайте, а если мы здесь сделаем так?» не важно что, и Глори говорит: «Отличная идея. Давайте так и сделаем».

Что было самым сложным?

МАРИЯ: Было очень сложно сделать, чтобы цвета волос и кожи менялись. Поэтому Нани нарисовала каждый вариант отдельно, а мы оставили для них место в коде. То есть нам не пришлось менять их средствами программы — они все уже были в наборе.

А что вам больше всего понравилось?

ГЛОРИ: Нам очень хотелось придать девочкам уверенности в себе, научить их любить себя, поэтому мы с радостью взялись за проект. Тем более что мы подруги и отлично провели время.

МАРИЯ: Невероятное ощущение — создать с нуля что-то действительно работающее, вдохнуть во что-то жизнь. До «Программирования для девочек» я ничего не знала о компьютерах, и когда в итоге я увидела, что моя идея так удачно воплощается в жизнь, превращается в игру, которой я очень-очень горжусь... это было потрясающе.

ЗАХРА: Я раньше никогда не писала программ. Было так здорово увидеть, что у нас все получилось! Когда мы стали проверять игру и она заработала как надо, я даже расплакалась. Я научилась программировать и познакомилась со своими лучшими подругами.



КОМПЬЮТЕРНЫЕ ИГРЫ

Ну как? Появились идеи?



МНЕ НРАВИТСЯ, ЧТО ОНИ С ПОМОЩЬЮ ИГРЫ РЕШИЛИ ПОКАЗАТЬ ДЕВОЧКАМ, ЧТО ЛЮБАЯ ИЗ НИХ КРАСИВА И СПОСОБНА ПРЕУСПЕТЬ В ЛЮБОЙ ПРОФЕССИИ.

МНЕ ТОЖЕ! И У МЕНЯ ИДЕЯ. Я ЛЮБЛЮ ЖИВОТНЫХ И СОСТОЮ В НЕСКОЛЬКИХ ПРИРОДООХРАННЫХ ГРУППАХ, КОТОРЫЕ БОРЮТСЯ ЗА СОХРАНЕНИЕ ВЫМИРАЮЩИХ ВИДОВ. НАВЕРНОЕ, ИГРА — ХОРОШИЙ СПОСОБ ЗАИНТЕРЕСОВАТЬ ЭТИМ И ДРУГИХ.

ДА, МОЖНО СДЕЛАТЬ ИГРУ, ГДЕ НУЖНО БУДЕТ ПОМОЧЬ СОХРАНИТЬ СРЕДУ ОБИТАНИЯ РАЗНЫХ ВИДОВ В ДИКОЙ ПРИРОДЕ. ТАК ИГРОКИ УЗНАЮТ, ПОЧЕМУ ЭТИ ВИДЫ ВЫМИРАЮТ И КАК ИХ СПАСТИ.



Давай вспомним, чему мы научились, и составим план.
С чего мы начинаем?

С исследования! Надо погуглить вымирающие виды и узнать, где они обитают. А потом решим, какие из них нам взять.

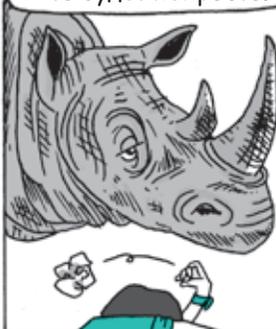
Да, и надо посмотреть, нет ли других игр про вымирающих животных.

Так, пора начать собирать симпатичные фотографии с животными.



А как мы будем рисовать места их обитания? Надо, чтобы они выглядели в точности как в жизни?

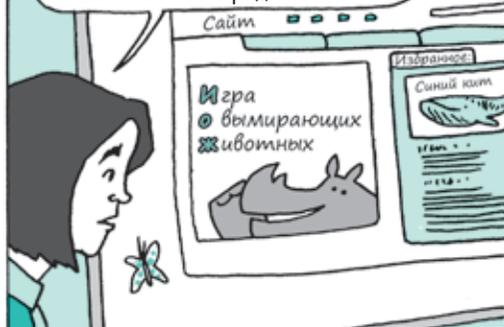
А сами животные? Это будет непросто.



Может, лучше сделать упрощенную версию с мультяшными изображениями животных?

Или разместить игру на сайте, где будут странички с фотографиями и информацией о реальных животных и их среде обитания.

Отличная идея. Тогда сделаем игру на Scratch, а сайт, наверное, на HTML.



КОМПЬЮТЕРНЫЕ ИГРЫ

Молодцы, девочки. Теперь давайте подумаем, как зарабатывать очки для победы или проигрыша в игре. Какие у нас будут переменные, циклы и условные конструкции?

ПУСТЬ БУДЕТ ПЕРЕМЕННАЯ
ДЛЯ ЧИСЛА ЖИВОТНЫХ,
КОТОРЫХ ИГРОК СПАС!

```
переменная totalAnimalsSaved = 15;
```



В КОНЦЕ ИГРЫ МОЖНО
ИСПОЛЬЗОВАТЬ УСЛОВНУЮ
КОНСТРУКЦИЮ, ЧТОБЫ
ОПРЕДЕЛИТЬ, ВСЕХ ЖИВОТНЫХ
ИГРОК СПАС ИЛИ НЕТ.



```
if (totalAnimalsSaved == 15) {  
    alert("Вы спасли всех животных!");  
}
```

Как видишь, разработка игры потребует времени и сил. Но если оставаться открытой для новых возможностей, даже самая простая игра может повлиять на мир так, как ты и не мечтала. Послушай, что говорит об этом Челси Хауи, разработчица компьютерных игр.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

СОВЕТЫ ПРОФЕССИОНАЛОВ РАЗРАБОТЧИК КОМПЬЮТЕРНЫХ ИГР



Я пошла работать в игровую индустрию, потому что всю жизнь играла и мечтала создавать миры, персонажей и истории, в которые смогут окунуться другие. Когда я узнала, что можно делать игры и получать за это деньги, я поняла: вот чем я хочу заниматься.

Но осуществить мечту оказалось непросто: Челси пришлось самой прокладывать дорогу.

В колледже я решила изучать языки, потому что большинство моих любимых пи-

сателей изучали языки и лингвистику. Оказалось, что в моем колледже была программа, посвященная играм, но там было всего два предмета. Поэтому я выбрала другую программу, чтобы самой построить свою специальность вокруг разработки игр.

У меня был потрясающий профессор — он понял, что меня интересует слияние технологий и искусства, и помог мне их совместить.

Получив работу, Челси обрела возможность создавать игры, о которых мечтала еще ребенком.

Я начала делать игры еще в колледже. На одну из них издатель подписал со мной контракт, и сразу после колледжа мы с другом основали собственную компанию. В то же время я работала продюсером, делала игры для Nintendo Wii, когда это еще была совсем новая, модная штука. Я занималась Farmville, симулятором фермера для соцсетей, еще когда в него каждый день играли 32 миллиона человек. Это

КОМПЬЮТЕРНЫЕ ИГРЫ

было невероятно — увлечь столько людей игрой, где нужно работать на ферме и заботиться о животных и растениях. В детстве я видела много довольно жестоких игр, поэтому было здорово трудиться над игрой, в которую бабушки играли вместе с внуками. Люди всех возрастов со всего мира полюбили нашу игру.

И оказалось, что игры — это не просто развлечение; они способны изменить нашу жизнь.

Я занималась игрой SuperBetter, где мы с помощью психологии и игровых элементов помогали людям преодолевать трудности. Когда мы изучали психологию игр, то узнали потрясающую вещь: когда люди играют в игры, они знают, что могут выиграть, поэтому пытаются снова и снова. Ты садишься играть с таким настроем: у меня получится, надо еще потренироваться, я знаю, что могу победить. И поэтому стараешься изо всех сил. Мы взяли этот настрой и применили его к жизненным проблемам: если не опускать руки, все у тебя получится.

Итак, каким же будет наш следующий проект?



Подскажу: программирование — это не только игры и развлечения. Оно может стать искусством, оживить персонажей комикса или превратить улыбку в песню.

МУЛЬТИПЛИКАЦИЯ



ФОТОСЪЕМКА



МОДА



МУЗЫКА





ЦИФРОВОЕ ИСКУССТВО И ДИЗАЙН

НАВЕРНЯКА У ТЕБЯ, ПОМИМО КОМПЬЮТЕРОВ, ЕСТЬ И ДРУГИЕ ХОББИ; но чем бы ты ни увлекалась, программирование может тебе в этом помочь. Девочки, вы бы хотели применить программирование в своем любимом занятии?

Я БЫ ХОТЕЛА УЗНАТЬ, КАК ИСПОЛЬЗОВАТЬ ПРОГРАММИРОВАНИЕ В АРТ-ПРОЕКТАХ. У МЕНЯ УЖЕ ЕСТЬ PHOTOSHOP И ILLUSTRATOR, НО МОЖНО ЛИ НАПИСАТЬ КОД ДЛЯ РИСУНКА?



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

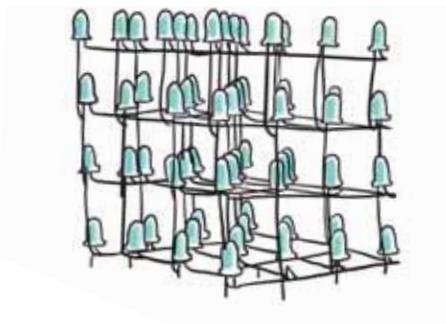
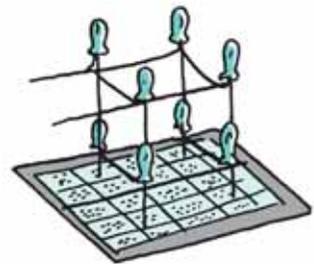
Разумеется! Программирование пригодится в любой области искусства, будь то изобразительное искусство, дизайн, мода, музыка, сценические выступления, фото- и видеосъемка или мультипликация. Ведь, в конце концов, искусство — это создание из подручного материала (красок или глины, инструментов или звуков, движений или слов) чего-то совершенно нового. Код — это такой же материал, инструмент для творчества, и только ты сама устанавливаешь границы его применения.

Но не обязательно верить мне на слово. Послушай, что расскажет Трэн П. о том, как ее команда — Трэн, Сандра В., Анджела К. и Лили И. — с курсов «Программирования для девочек» в Сан-Франциско превратила набор светодиодных ламп в интерактивную систему освещения, мерцающую в такт музыке. Идеальная находка для вечеринок и танцев, система «Ярче звук» превращает музыку в свет — или, как говорит Трэн, «если в музыке мы слышим наши чувства, то в этом светодиодном видеокубе можно увидеть музыку».

eee) Ярче звук (eee

Расскажите, чем вы вдохновлялись, когда придумывали «Ярче звук».

ТРЭН: Мы случайно оказались в одной команде, разговорились, и выяснилось, что мы



все любим музыку, рисование и дизайн.

Мы говорили о том, что на танцах в школе освещение — это всегда старомодные диско-шары и стробоскопы, и было бы неплохо придумать что-нибудь получше. Поэтому мы решили сделать световой куб, который мигал бы в такт музыке.

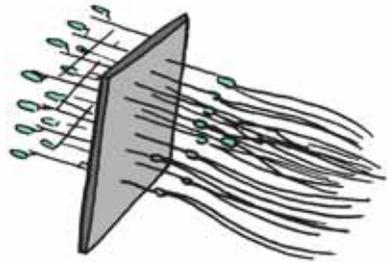
ЦИФРОВОЕ ИСКУССТВО И ДИЗАЙН

Ого, какая необычная идея! И с чего же вы начали?

Мы погуглили другие проекты со светодиодными лампами и нашли несколько полезных инструкций, но они были в основном для кубов со стороной 20 сантиметров. У нас была всего неделя, поэтому мы решили сделать куб со стороной 10 сантиметров. Мы взяли платформу Arduino, а код написали на языке Processing.

Где вы брали нужные материалы: лампы, оборудование?

В «Программировании для девочек» были лампы и платформа, оставалось их только соединить. Мы взяли паяльник и провода. Сгибали провода, спаивали их друг с другом, а потом проверяли каждую лампочку, горит ли она, и каждый ряд.

*Как вы писали программу для этого проекта?*

Я занималась оборудованием, а код писали остальные. Сначала они смотрели, какое у разных песен число ударов в минуту¹¹, а потом на основании этих чисел составляли схемы подсветки. Мы сделали музыкальный файл, чтобы определять число ударов в минуту в песне. Как только программа понимает ритм, она выбирает подходящую схему подсветки.

Что было самым интересным?

Интереснее всего было подсоединять лампы к платформе и паять. Иногда лампочка ломалась и нужно было очень осторожно ее заменять. Но мне понравилось подключать их к платформе и работать с паяльником. Раньше я ничего подобного не делала, а когда учишься чему-то новому, чувствуешь такое удовлетворение!

¹¹ Удары в минуту (англ. beats per minute, BPM) — скорость воспроизведения музыкальной композиции. Определяется числом четвертных нот в минуту. — *Примеч. пер.*

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Ваш проект показал, что программирование — это тоже творчество.

Я много рисую, поэтому с удовольствием взялась за проект, где визуальная часть играет большую роль. Еще меня интересует виртуальная реальность и то, что она может принести в изобразительное искусство. Я советую всем, кто любит рисовать или занимался дизайном, научиться программировать. Зачем упускать возможности, когда их так много? Ведь всегда можно придумать, как соединить программирование со своим любимым занятием.



«Ярче звук» — всего лишь один пример того, как можно применить программирование в изобразительном искусстве, дизайне и музыке. Прежде чем начинать мозговой штурм, давай посмотрим, что это могут быть за применения.

ИЗОБРАЗИТЕЛЬНОЕ ИСКУССТВО

Сейчас искусство и программирование сплетаются все теснее — от инноваций в мультипликации, кино и фотосъемке до трехмерных пейзажей в виртуальной реальности и сцен в интерактивных спектаклях, запрограммированных, чтобы изменяться от одного прикосновения. Последнее слово этой тенденции — зарождающееся **ПРОЦЕДУРАЛЬНОЕ ИСКУССТВО**. Художники не выбирают, какой формы, стиля и цвета будет изображение, — вместо этого они создают алгоритм, позволяя компьютеру творить в заданных рамках.

Программирование способно оживлять статичные изображения. Сейчас появились комиксы и картины, которые начинают двигаться, если навести на них камеру телефона. Технологии захвата движения, высококачественное сканирование и трехмерное моделирование помогают художникам создавать невероятно детальные, бесшовные анимированные миры для полнометражных и короткометражных фильмов.

ЦИФРОВОЕ ИСКУССТВО И ДИЗАЙН

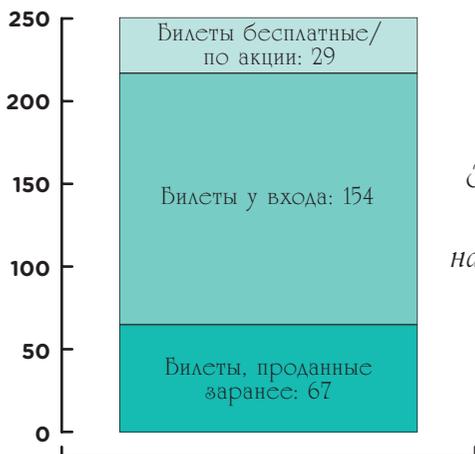
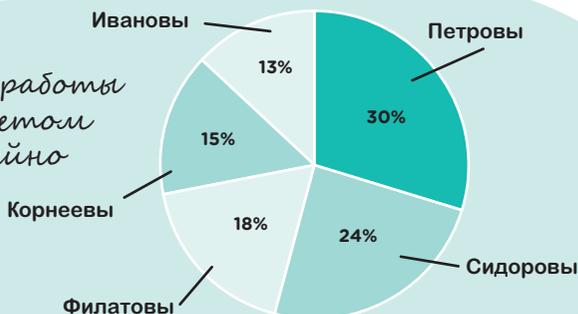
Еще одна популярная область программирования — визуализация данных: художники используют цвета и формы, чтобы иллюстрировать графики и диаграммы и помочь нам лучше понять их смысл.



*Голосование:
куда кружку
естественных
наук
отправиться
на экскурсию*



*Доход от работы
няней летом
посемейно*



*Продажи
билетов
на школьный
мюзика*



МОДА И ДИЗАЙН

В моде не последнюю роль играют умная одежда и аксессуары, сочетающие функциональность со стилем. Очки, украшения и одежда оснащаются процессорами, способными передавать данные о твоём местонахождении, чтобы защитить тебя. Существует спортивная одежда, которая отслеживает сердцебиение и пройденную дистанцию, заботясь о твоём здоровье. Большинство функций твоего телефона можно перепоручить стильным часам или паре очков. Есть даже платья со светодиодами — в таком наряде можно блистать на любой вечеринке! И все это стало возможным благодаря программированию.

Изобретение трехмерной печати

открыло новые возможности и для дизайнеров. Трехмерный принтер выдавливает расплавленный материал, который укладывается сверхтонкими слоями, кусочек за кусочком. Слои накладываются один на другой и создают очертания объекта. Материал застывает, и получается готовое изделие. Обычно

в качестве материала используют пластик, но инженеры шоколадного магната Hershey's научились делать потрясающие скульптуры из шоколада! Благодаря трехмерной печати дизайнеры смогли создавать тщательно проработанные нетяжелые объекты, превращая абстрактный код во вполне материальные предметы.



МУЗЫКА

Между созданием музыки и написанием кода не такая уж большая разница, ведь и то и другое — алгоритмы. Нотное письмо — это способ

ЦИФРОВОЕ ИСКУССТВО И ДИЗАЙН

закодировать музыку. Нет ничего удивительного в том, что существует целая область музыки, посвященная **АЛГОРИТМИЧЕСКОЙ КОМПОЗИЦИИ**. Так же, как в процедуральном искусстве, в алгоритмической композиции компьютер пишет музыку, следуя алгоритму, без вмешательства композитора-человека. В качестве данных для этих алгоритмов используют фразы из книг, черты лица или отметки на карте: можно превратить ге-

ниальный роман в песню или создать симфонию своей улыбки. Продаются комплекты аппаратного обеспечения, с помощью которых можно любой предмет превратить в электронный музыкальный инструмент. Только представь: сыграть на пончике с повидлом!



**Я МОГУ СТОЛЬКО
ВСЕГО СДЕЛАТЬ!**

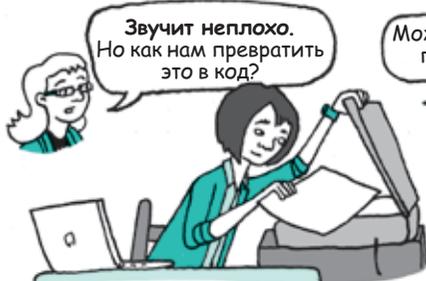
**МОГУ АНИМИРОВАТЬ
СВОИ КОЛЛАЖИ, СДЕЛАТЬ
ИХ ИНТЕРАКТИВНЫМИ ИЛИ
НАПИСАТЬ ПРОГРАММУ, ЧТОБЫ
КОМПЬЮТЕР САМ СОСТАВЛЯЛ ИХ
ИЗ МОИХ ПОДБОРОК!**



Да, ты все это можешь. Поэтому пора нам остановиться на чем-то одном и набросать план.



Обычно я вырезаю картинку из журналов или рекламных проспектов и наклеиваю их на бумагу.



Можно отсканировать готовый коллаж

и с помощью Photoshop разделить его на части.



ЦИФРОВОЕ ИСКУССТВО И ДИЗАЙН

Отличный мозговой штурм! Вы уже начинаете основательно подходить к делу. Теперь надо решить, как и при каких условиях части коллажей будут двигаться, и можно приступать к псевдокоду.

Я ХОЧУ ИСПОЛЬЗОВАТЬ ЦИКЛ,
ЧТОБЫ МОЕ ЛИЦО КРУТИЛОСЬ,
ЕСЛИ НА НЕГО НАЖАТЬ!

если изображение_лицо нажато:
повернуть изображение_лицо
на 360 градусов



В МОЕМ КОЛЛАЖЕ ЕСТЬ КАРТИНКА
СО ЩЕНКОМ. Я ХОЧУ, ЧТОБЫ ЩЕНОК
ВСЕ ВРЕМЯ РОС!



повторять бесконечно:
увеличить ширину изображение_щенок
на 5 пикселей
увеличить высоту изображение_щенок
на 5 пикселей

В написании программы, как и в любом искусстве, нужно быть готовой к экспериментам и к движению вперед методом проб и ошибок. Но даже если все идет не по плану, ты все равно можешь добиться потрясающих результатов. Спроси Даниэль Фейнберг.

СОВЕТЫ ПРОФЕССИОНАЛОВ ОПЕРАТОР-ПОСТАНОВЩИК PIXAR



Наверняка ты знаешь и любишь такие мультфильмы, как «В поисках Немо», «Приключения Флика», «Корпорация монстров», «ВАЛЛ-И» и «Храбрая сердцем». Над ними работала Даниэль Фейнберг. Ее работа — это удивительный сплав программирования и искусства. Сегодня она расскажет нам о том, с чего начинала, что больше всего любит в своем деле, и о том, как из ошибки может родиться находка!

Моя должность называется «оператор-постановщик». Я отвечаю за освещение в фильмах. Мы создаем в компьютере трехмерный мир, а я расставляю по местам источники освещения. Если нужен закат, я создаю источник освещения — солнце, помещаю его над горизонтом, делаю оранжевым и получаю замечательный пурпурно-синий цвет, отражающийся от неба. И я могу управлять тенью, цветом, атмосферой и свойствами света.

В работе Даниэль использует программирование для самых разных целей.

Программы, которые мы используем, чтобы расставить источники освещения, содержат миллионы миллионов строчек кода. Иногда мы сами пишем небольшие программы, чтобы облегчить себе жизнь, или с помощью кода создаем новые источники освещения. Для мультфильма «В поисках Немо» я работала над таким источником, мы назвали его «Мрачный свет». Он передает, как свет преломляется в воде.

Даниэль полюбила программирование и искусство еще в детстве: она начала изучать программирование в четвертом классе.

ЦИФРОВОЕ ИСКУССТВО И ДИЗАЙН

Первым моим опытом программирования было создание изображений.

Я была очарована: пишешь код, а получаешь картинку.

Мои родители — творческие люди, сестра изучала в колледже изящные искусства.

Нас с детства записывали на уроки рисования. В подвале у нас был огромный стол, а на нем — все принадлежности для рисования, какие только можно пожелать. Мы с сестрой спускались в подвал и часами рисовали и наслаждались.

Даниэль считает, что рисование и написание программ во многом похожи.

Написание кода — это вполне творческий процесс. Мне нравится мысль, что у тебя есть набор команд на языке, который ты используешь. Перед тобой цель, которой нужно достичь, и ты разбиваешь ее на более мелкие задачи, но в твоём распоряжении — конечное число команд. Чем не творчество — искать способ, как с помощью этого набора добиться своей цели?

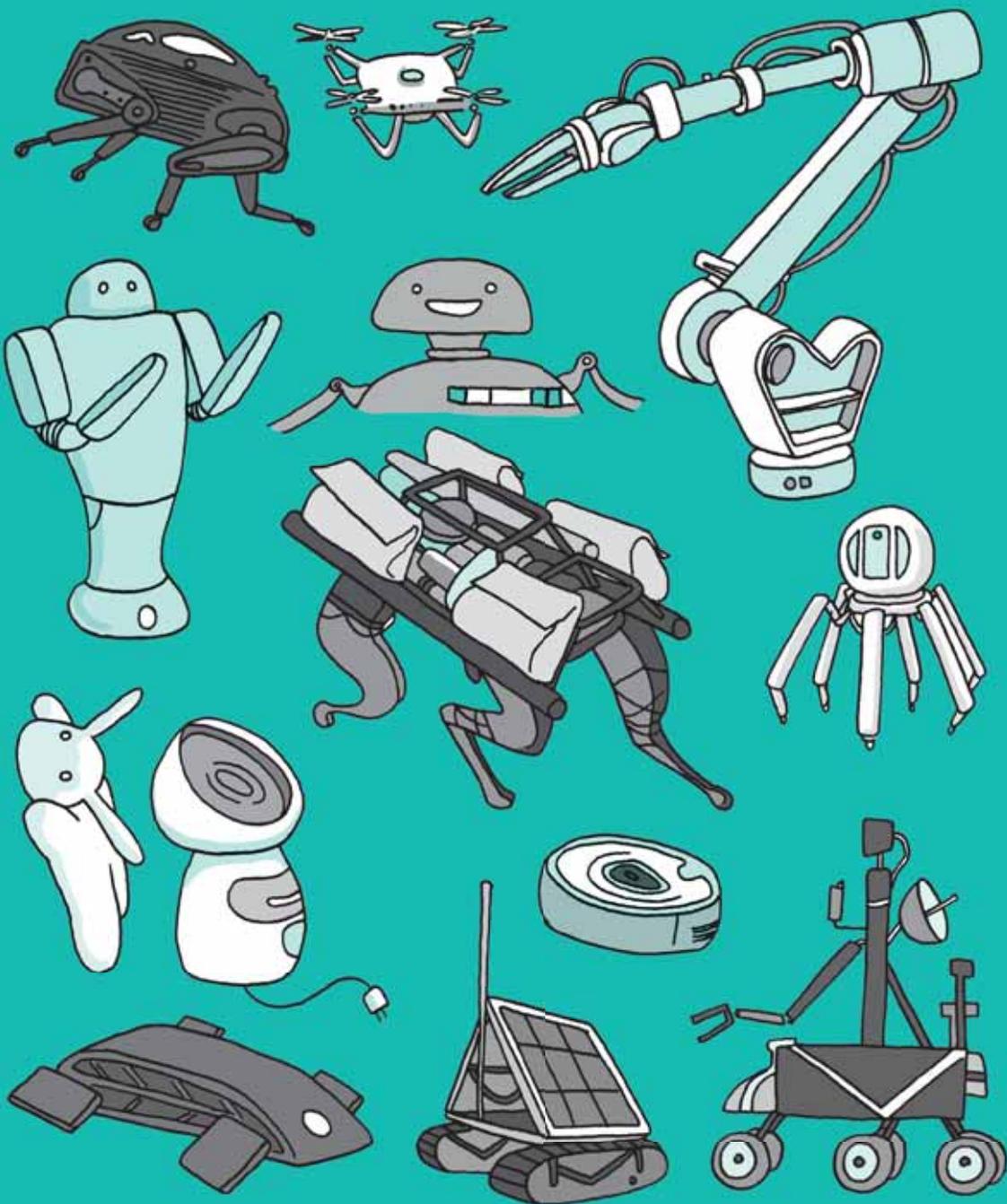
Даже если твоя идея или программа не работает так, как было задумано, она может подарить тебе замечательную находку — то, что люди искусства называют счастливой случайностью.

Большинство случайностей в программировании счастливыми не назовешь.

Но в самом начале работы над «Храброй сердцем» я делала лесной пейзаж с освещением и туманом. В моей программе была ошибка, и освещение исчезло. Остался только туман, все остальное было черным. От этого лес приобрел глубину, а растения превратились в интересные силуэты. Это было очень красиво, но я никогда бы этого не увидела, если бы компьютер не ругнулся на программу и не выдал такую картинку. Поэтому очень важно быть открытым для вдохновения, когда оно приходит.



Быть открытым для вдохновения и новых возможностей — это важное условие того, чтобы стать успешным программистом. Важное, но не единственное. Если прислушиваться к мнениям других людей, твой проект может обрести новую силу и в самом деле изменить чью-то жизнь.





РОБОТЫ

ИТАК, МЫ ОБСУДИЛИ КОМПЬЮТЕРНЫЕ ИГРЫ, ЦИФРОВОЕ ИСКУССТВО и дизайн. О какой еще области программирования вы хотели бы поговорить?

**ПРО РОБОТОВ,
ДАВАЙТЕ ПРО РОБОТОВ!**
Я БЕЗ УМА ОТ РОБОТОВ, ОСОБЕННО
ТЕХ, ЧТО ИССЛЕДУЮТ КОСМОС,
КАК МАРСОХОДЬИ!



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Не могу обещать, что к концу этой главы мы сможем сделать марсоход, но существует немало очень важных проектов, связанных с роботами.

Эти невероятные машины веками вдохновляли нас. Когда речь идет о роботах, большинство из вас наверняка представляет себе человекоподобного дроида, похожего на персонажа из «Звездных войн», — механического человека, способного думать, чувствовать, поддерживать разговор и двигаться. Но роботы бывают всевозможных форм и размеров и выполняют самые разные задачи. И все роботы до единого запрограммированы.



Они помогают людям во многих областях. Бывают производственные и сельскохозяйственные роботы, способные выполнять слишком тяжелую, опасную или монотонную работу. Не всякий человек захочет заниматься таким тяжелым трудом.

Есть и медицинские роботы: от электронных хирургических инструментов, с помощью которых хирурги делают операции, требующие предельной точности, до дозаторов таблеток, которые ездят за медсестрами по больницам и выдают лекарства. Сейчас инженеры создают роботов-докторов с экранами и сенсорами: они помогут докторам-людям связываться с пациентами в отдаленных районах и лечить их.

Механические конечности, протезы и помощники изменяют жизнь инвалидов и хронически больных людей, возвращая им контроль над утраченными, поврежденными или ослабленными частями тела.

Не забудем и о нанороботах: хотя сейчас они существуют только в научной фантастике, ученые и инженеры работают не покладая рук, чтобы эти

РОБОТЫ

крошечные — не крупнее песчинки! — роботы стали реальностью. Есть надежда, что в будущем нанороботы смогут проникать в человеческое тело, чтобы выявлять и лечить болезни без хирургического вмешательства. Инженеры также ищут способ запрограммировать нанороботов объединяться в рои (как у насекомых) для совместной работы или ремонта в труднодоступной местности.

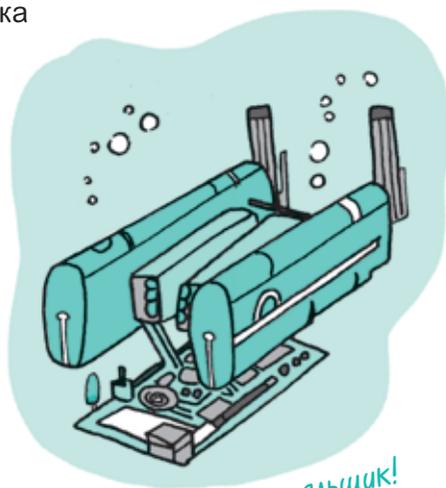


Наноробот (увеличенный в 25 миллионов раз!)

Существуют также дроны и дистанционно управляемые роботы, незаменимые в труднопроходимой и опасной местности. Они помогают сотрудникам правопорядка, военным и поисково-спасательным командам в зонах боевых действий и при ликвидации последствий стихийных бедствий.

Похожих роботов используют ученые, чтобы исследовать слишком далекие или опасные для человека

районы. Их роботы ползают по арктическим ледникам, странствуют по дальним пределам нашей Солнечной системы и погружаются в океанские глубины. Благодаря данным, которые собирают эти роботы, и изображениям, которые они присылают, мы открываем для себя новые стороны нашей Вселенной.



Робот-ныряльщик!



НО ЭТО БОЛЬШИЕ, СУПЕРМОЩНЫЕ РОБОТЫ, КОТОРЫЕ УМЕЮТ ДЕЛАТЬ ПОТряСАЮЩИЕ ШТУКИ. А КАК СДЕЛАТЬ РОБОТА, КОТОРЫЙ ПРИГОДИТСЯ В ОБЫЧНОЙ ЖИЗНИ?

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Подумай о том, чем ты занимаешься каждый день. Как робот может помочь тебе? И не обязательно только тебе, ведь твой робот может помогать и другим. Именно такого робота придумали четыре юные программистки: Эмбер С., Эмили Д., Идализ Д. и Янаншали С.

eee Робот-поводырь *eee*

Как вы придумали своего робота-поводыря?

ЭМБЕР: В «Программировании для девочек» мы делали танцующего робота. Это было очень весело, поэтому, когда пришло время выбирать собственный проект, мы решили, что будем делать робота. Но мы хотели, чтобы он не просто танцевал, а помогал людям.

Расскажите, как он работает.

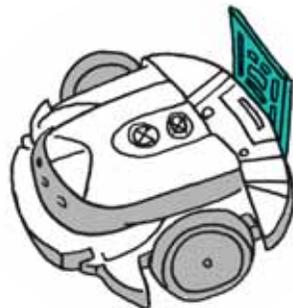
ЯНАНШАЛИ: Прототипом был маленький робот с датчиками движения и инфракрасного излучения. Он может просканировать местность, обнаружить находящиеся перед ним объекты и рассчитать путь, чтобы безопасно обогнуть их. Мы запрограммировали его так, чтобы он издавал различные звуки, когда приближается к препятствию. Тогда пользователь поймет, что перед ним что-то находится.

Он у вас заработал?

ЯНАНШАЛИ: Да, он научился передвигаться, замечать объекты, останавливаться и огибать их. Мы построили лабиринт из коробок, и он его прошел.

Чему научил вас этот проект?

ЭМБЕР: Самый главный вопрос — сколько должно быть сенсоров. В следующий раз мы добавили бы больше сенсоров: эта версия получилась немного неповоротливой и не могла преодолевать

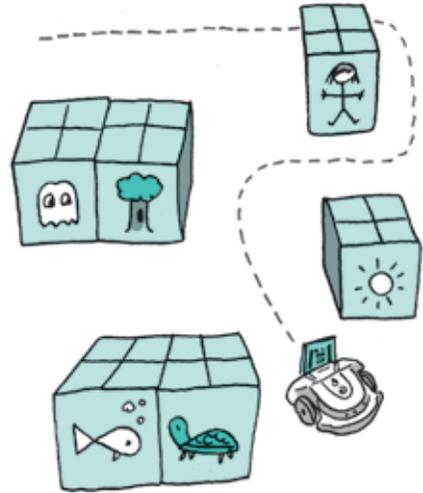


РОБОТЫ

пороги и лестницы. Но в роботостроении интереснее всего то, что ты прямо в процессе видишь, как работает устройство. Ты создаешь его и одновременно тестируешь, исправляешь ошибки и сразу что-то меняешь.

Что вам больше всего понравилось?

ЯНАНШАЛИ: Работать вместе было очень полезно. Нас было четверо, а это четыре разных образа мыслей. Каждая привнесла в проект свои сильные стороны и свою точку зрения, поэтому мы могли обсуждать наши идеи и помогать друг другу с разными вопросами. И это было очень весело.



Ну, что думаете? Все еще хотите построить робота — космического исследователя или появились идеи для чего-то более понятного и близкого?

Я ПОДУМАЛА О СВОЕМ МЛАДШЕМ БРАТЕ. ОН УЧИТСЯ ВО ВТОРОМ КЛАССЕ. Я ХОЧУ СДЕЛАТЬ РОБОТА, КОТОРЫЙ ПОМОЖЕТ ЕМУ С ДОМАШНИМИ ЗАДАНИЯМИ. БУДЕТ УСТРАИВАТЬ НЕБОЛЬШИЕ ОПРОСЫ, ПОМОГАТЬ РЕШАТЬ ЗАДАЧИ И В ТО ЖЕ ВРЕМЯ СОСТАВЛЯТЬ БРАТУ КОМПЬЮТЕРНУЮ ПОМОЩЬ, ЧТОБЫ ОН НЕ ОТВЛЕКАЛСЯ.



Это очень заботливо с твоей стороны. Прекрасная идея. Так как же нам ее осуществить?



РОБОТЫ

ВЫБЕРЕМ СЛУЧАЙНЫЙ
ВОПРОС, И ПУСТЬ РОБОТ
ЕГО ПРОИЗНЕСЕТ!



Из списка вопросов случайным
образом выбрать один и сохранить
в переменной
Робот произносит текст вопроса

А ПОТОМ ПУСТЬ
ПОДОЖДЕТ НЕСКОЛЬКО
СЕКУНД И ДАСТ ПРАВИЛЬНЫЙ
ОТВЕТ.



Подождать 5 секунд
Робот произносит ответ на вопрос

Твоего брата ждет незабываемый сюрприз. Поверь специалисту по робототехнике Аяне Ховард: нет ничего лучше того момента, когда твой робот оживает.

СОВЕТЫ ПРОФЕССИОНАЛОВ РОБОТОТЕХНИК И ПРОФЕССОР ТЕХНОЛОГИЧЕСКОГО ИНСТИТУТА ДЖОРДЖИИ



Аянна Ховард

В робототехнике Аянне Ховард нет равных. Вместе с NASA она создавала космических роботов. Ее роботы исследовали ледяные пустоши Антарктиды. Мы спросили ее о том, как она заинтересовалась робототехникой, что входит в понятие «робот» и как взгляд на мир с другой стороны повлиял на ее работу.

Мне нравится робототехника, потому что тут сразу видишь, работает твоя программа как следует или нет. Если ошибешься с переменной, робот врежется в стену.

В робототехнике можно использовать разные способы познания. Я могу наблюдать, как робот двигается, или потрогать его, передвинуть и запрограммировать. Люди боятся, что, если они станут программистами, им придется целый день сидеть за монитором, но с роботами нужно взаимодействовать, нужно двигаться.

Аянна начинала с создания роботов для NASA. Тогда она поняла, как полезно взглянуть на свое изделие с точки зрения пользователя.

Нам досталась миссия по исследованию Марса. Хотя это была космическая робототехника и робототехника для опасной местности, я сосредоточилась на том, для чего нам вообще понадобились роботы. На научных исследованиях. Мне нужно было понять, как думают ученые, как бы они ориентировались в этой местности и исследовали ее. Когда я пришла в Технологический институт Джорджии, мы делали роботов для дру-

РОБОТЫ

гих исследований. наших роботов отправляли не в космос, а во льды и на дно океана. Они помогали ученым понять, почему тают ледники и что происходит при глобальном потеплении. Я работала не только с компьютерными специалистами, но и с климатологами и микробиологами.

Привычка учиться у других привела Аянну к тому, чем она занимается сейчас.

Меня очень интересует применение роботов в медицине, например помощь детям-инвалидам. Идея возникла случайно. Я работала в летних лагерях, посвященных робототехнике, и у меня в группе была девочка с нарушением зрения. Наше оборудование не было на нее рассчитано, но она была очень умна и справилась. И я поняла: нужно создать условия, в которых у всех детей будут одинаковые шансы достичь успеха. Я стала работать сначала с детьми с нарушениями зрения, а потом — с двигательной недостаточностью.

Над каким бы проектом ни работали ее ученики, Аянна всегда старается напомнить им, как ценны их навыки программирования.

Программирование — это не просто нули и единицы на экране. Когда ты пишешь программу для больничной системы, ты спасаешь чью-то жизнь. Когда ты создаешь систему программирования, которую будут изучать в школе, ты влияешь на поколения учеников. Твои нули и единицы дают тебе возможность воздействовать на мир вокруг и изменять его.



Не могу дождаться, когда увижу, как вы, девочки, измените мир, когда научитесь программировать!

Нам осталось обсудить еще одну важную тему. Этими программами мы пользуемся каждый день и самыми разными способами. На самом деле у вас уже есть ключ к тому, чтобы их улучшить, хотя вы, возможно, об этом даже не подозреваете.



10

САЙТЫ, МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ И ИНТЕРНЕТ- БЕЗОПАСНОСТЬ

НУ КАК, ДОГАДАЛИСЬ, КАКУЮ ЕЩЕ ВАЖНУЮ ОБЛАСТЬ программирования мы пока не обсудили?

Подсказка: большинство из вас пользуется этим постоянно. Возможно, даже прямо сейчас!

САЙТЫ
И МОБИЛЬНЫЕ
ПРИЛОЖЕНИЯ?



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Именно! Дизайн и разработка сайтов и мобильных приложений — один из лучших способов применить свои навыки программирования, не откладывая в долгий ящик. Вам понадобятся только компьютер и идея, и можно приниматься за творчество. Дети и подростки — очень активные пользователи приложений и онлайн-сервисов. Вы уже задаете тон на этом рынке. Вы не только определяете успех того или иного продукта, когда жмете «Нравится» и «Поделиться», но и обучаете своих родителей обращаться с техникой. Это значит, что во многом именно вы определяете, на какие сайты заходят ваши родные, какие приложения они покупают, используют и поддерживают. Как продвинутые пользователи, вы уже знаете, какие продукты вам нравятся. А теперь вы сможете еще и улучшать их!



Ого, я смотрю, процесс пошел. Об этом я и говорила. Вы используете технологии и точно знаете, что хотите и что вам нужно. Когда вы научитесь программировать, вам больше не придется ждать, пока кто-нибудь напишет нужное приложение, — можно написать его самим!

САЙТЫ, МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ И ИНТЕРНЕТ-БЕЗОПАСНОСТЬ

Но, когда делаешь приложения и сайты, нужно кое о чем помнить.

САЙТЫ

Есть множество способов создать сайт. Можно начать с готового шаблона или написать свой. В любом случае одним из важнейших элементов, который следует продумать при создании сайта, остается пользовательский интерфейс. Для чего служит этот сайт?

Кто будет им пользоваться? Какую информацию или услуги сайт будет предоставлять пользователям? Ответы на эти вопросы помогут решить, на каком языке написать сайт и какими свойствами его наделить.

Будут ли посетители смотреть на твоем сайте видео или фотографии? Как будет отображаться сайт, если зайти на него с телефона? Будут ли пользователи заказывать какие-нибудь товары? Если пользователи будут предоставлять персональные данные, как защитить эти данные от постороннего доступа?

Над этими вопросами стоит задуматься и при разработке приложений. Кроме того, там есть свои важные моменты.



ПРИЛОЖЕНИЯ

Приложения бывают разных видов — это важно, потому что разные приложения создаются по-разному.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

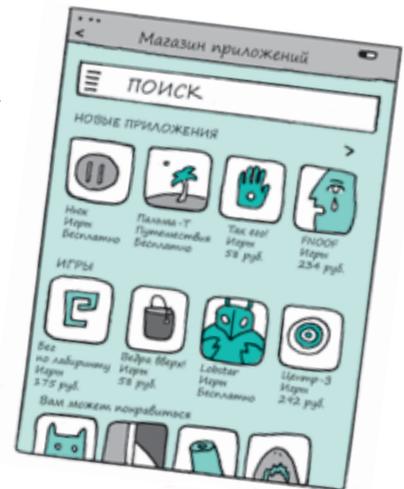
ВСТРОЕННЫЕ ПРИЛОЖЕНИЯ —

уже установлены на устройстве, живут на его начальном экране и открываются через иконки. Новые приложения можно найти в магазине приложений.

ВЕБ-ПРИЛОЖЕНИЯ ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ —

это сайты, которые выглядят и работают как приложения, но на самом деле запускаются через браузер. Получить доступ к ним можно через сайт, скачав его мобильную версию. У таких приложений есть ограничения: например, у них не всегда есть доступ к камере или к памяти устройства.

Приложения для устройств, работающих на операционных системах iOS и Android, пишут на разных языках. Если устройство работает на Android, то приложение нужно писать на Android App Inventor или Java, а если на устройстве установлена iOS, то для приложения выбирают Swift или Objective-C. Нелишним будет подумать, как ты собираешься распространять свое приложение, чтобы сразу знать, какой выбирать язык.



ВЫ ГОВОРИЛИ, ЧТО НУЖНО ЗАЩИЩАТЬ ДАННЫЕ ОТ ПОСТОРОННЕГО ДОСТУПА, КОГДА СОЗДАЕШЬ САЙТ. Я ВСЕГДА ДУМАЛА: КАК МЫ МОЖЕМ БЫТЬ УВЕРЕННЫ, ЧТО ДАННЫЕ, КОТОРЫЕ МЫ ЗАКЛАДЫВАЕМ В ПРОГРАММУ, ДА И САМА ПРОГРАММА НАДЕЖНО ЗАЩИЩЕНЫ?

Прекрасный вопрос.



БЕЗОПАСНОСТЬ И КОНФИДЕНЦИАЛЬНОСТЬ В ИНТЕРНЕТЕ

Чем большую роль в нашей жизни играет Интернет, тем важнее защищать свои персональные данные, финансовые реквизиты и даже домашние умные приборы (термостаты, охранные сигнализации и другие устройства) от атак злоумышленников. К сожалению, существуют хакеры, преступники и даже враждебно настроенные правительства и политические группировки, которые пользуются уязвимостями в **КИБЕРБЕЗОПАСНОСТИ**, чтобы красть информацию, деньги и отключать сервисы.

И ЧТО НАМ
ДЕЛАТЬ?



Сейчас активно развивается область информатики, которая специализируется на кибербезопасности и борется с киберпреступностью. Специалисты в этой области работают как в коммерческом секторе (с компаниями и клиентами), так и в правительственном (с государственными и федеральными службами). Они предоставляют следующие услуги.

- ★ **ШИФРОВАНИЕ ДАННЫХ** — создание кодов и паролей для защиты данных.
- ★ **ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ СЕТЕЙ** — поддержка сетей компании в рабочем состоянии, их своевременное обновление и чистка от вирусов и вредоносных программ.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

- ★ **ТЕСТИРОВАНИЕ И РАЗРАБОТКА КИБЕРЗАЩИТЫ** — тестовый взлом сетей, который позволяет найти уязвимости в системе безопасности, и разработка средств, чтобы устранить уязвимости.

Шифрование данных и поддержание интернет-безопасности — это программирование довольно-таки высокого уровня, но сейчас тебе нужно запомнить вот что: когда ты разрабатываешь сайт или приложение, всегда тщательно продумывай, какую информацию будут предоставлять твои пользователи и как ты будешь ее защищать. В Интернете есть множество ресурсов с полезными советами о том, как гарантировать сохранность данных и программ.

А Я ВСЕГДА ДУМАЛА, ЧТО СОБЛЮДАТЬ ИНТЕРНЕТ-БЕЗОПАСНОСТЬ — ЗНАЧИТ ПРОСТО ВЕСТИ СЕБЯ ОСТОРОЖНО В ИНТЕРНЕТЕ: НЕ ДАВАТЬ НЕЗНАКОМЦАМ СВОЙ ДОМАШНИЙ АДРЕС И НЕ НАЖИМАТЬ НА ПОДОЗРИТЕЛЬНЫЕ ССЫЛКИ, С КОТОРЫХ КАЧАЮТСЯ ВИРУСЫ.



И это тоже. Уверена, взрослые уже говорили тебе, на что нужно обращать внимание в Интернете; вот пара советов.

ЛИЧНАЯ БЕЗОПАСНОСТЬ ОНЛАЙН

Ты, наверное, уже поняла, что жить в нашем мире не так-то просто, особенно молодым девушкам. Онлайн-мир в этом смысле ничем не отличается: там обитают те же люди, что и в реальной жизни. Многие хотят познакомиться с кем-то, помочь другому, найти друзей и создать свое сообщество. Но есть и такие,

САЙТЫ, МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ И ИНТЕРНЕТ-БЕЗОПАСНОСТЬ

что скрывают свое настоящее лицо, чтобы обманывать, грабить, высмеивать, троллить незащищенных людей, а иногда и причинять им вред. Цифровой мир дарит невиданную свободу исследований и общения, но в Интернете понять, с кем ты общаешься, намного сложнее, чем в реальной жизни.

НЕ ТРУСИХА, А ОСТОРОЖНАЯ

Интернета не нужно бояться, но следует соблюдать осторожность, особенно в том, что касается персональных данных. Если кто-то хочет увидиться с тобой лично, убедись, что он не имеет ничего против того, чтобы встретиться сначала с твоими родными или опекунами. Точка. Если человек откажется — это тревожный звонок. О таком случае необходимо рассказать взрослому, которому можно доверять. Ты же не сядешь в машину к незнакомцу с улицы? Точно так же не соглашайся на личную встречу и не давай свои персональные данные незнакомому человеку из Интернета, каким бы милым он ни казался.

Селеста Б., Джули П., Ямин Л. и Анни Х., участницы «Программирования для девочек» в Сизтле, не хотели просто сидеть и беспокоиться без толку. Чтобы помочь молодым девушкам и обеспечить им личную безопасность, они создали замечательное приложение «Ангел-хранитель».

eee) Ангел-хранитель (eee

Как возникла эта замечательная идея и для чего вообще служит это приложение?

ДЖУЛИ: Мы хотели сделать что-то такое, что пригодится людям и будет им по-настоящему полезно, а не просто развлечет. Поэтому мы решили, что сделаем что-нибудь, что нужно всегда.



ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

На рынке есть много приложений для безопасности, но мы не хотели, чтобы наше приложение просто с кем-то связывалось или отслеживало, где ты находишься. Нам нужен был универсальный ресурс безопасности. Так родился «Ангел-хранитель».

Расскажите, как работает приложение? Что оно дает?

СЕЛЕСТА: Когда открываешь приложение, можно включить режим безопасности, чтобы оно отслеживало, где ты находишься. Если нажать аварийную кнопку, то всем безопасным контактам (группе, которую ты сама назначаешь)

отправится сообщение с отметкой твоего местонахождения на картах Google. У «Ангела-хранителя» есть собственная база данных. Как только твои безопасные контакты получают сообщение, «Ангел-хранитель» разошлет им инструкции, как до тебя добраться. Кроме того, «Ангел» умеет находить поблизости работающие заведения, где можно укрыться, и дает инструкции, как туда добраться.

ДЖУЛИ: Еще есть страничка с подсказками, как избежать опасных ситуаций.

Мы хотели, чтобы наше приложение было полезно всем, но, наверное, особенно оно пригодится тем, кто только начинает жить один и не чувствует себя уверенно в большом городе или на ночных улицах.

Вы решили, что вам самим пригодится такое приложение, или писали его для кого-то еще?

СЕЛЕСТА: Мы все согласились, что неплохо иметь такое приложение. Все мы попадали в ситуацию, когда ты идешь где-нибудь одна, тебе страшно и хочется, чтобы кто-нибудь присматривал за тобой, но при этом не торчал у тебя за спиной. И все наши одноклассники и друзья говорили, что такое приложение нужно.



САЙТЫ, МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ И ИНТЕРНЕТ-БЕЗОПАСНОСТЬ

ДЖУЛИ: Наш девиз: «Позаботься о своей безопасности сам». Мы хотели, чтобы пользователь мог контролировать, когда и кому предоставить информацию о своем местонахождении. Ты всегда контролируешь то, что происходит вокруг тебя. Наше приложение — это не просто «ой, мне страшно, сейчас я нажму эту кнопку и буду ждать, пока кто-нибудь спасет меня». Оно дает тебе возможность действовать самой.

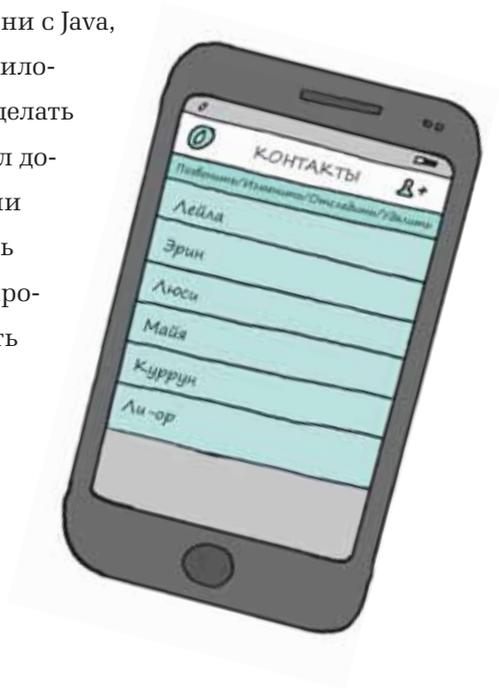
Как вы создавали приложение? Какой язык использовали?

СЕЛЕСТА: Никто из нас раньше не работал ни с Java, ни со Swift, а они нужны, когда делаешь приложение для Android или iOS. Мы не могли сделать веб-приложение, потому что нам нужен был доступ к сообщениям, контактам и геолокации на смартфоне. Поэтому мы воспользовались MIT App Inventor — это визуальный язык программирования, на нем очень просто писать приложения для Android.

ДЖУЛИ: Это как Scratch, только для приложений.

Вам понравилось работать в группе?

ДЖУЛИ: Это было очень здорово, потому что нам постоянно нужно было работать вместе. App Inventor — не самая интуитивно понятная программа, поэтому нам пришлось приложить усилия, чтобы достичь четкой цели. И процесс пошел: мы научились распределять задачи и в то же время работать в группе.



Ну как? Готовы нарисовать схему своего приложения?



САЙТЫ, МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ И ИНТЕРНЕТ-БЕЗОПАСНОСТЬ

НУЖНО УБЕДИТЬСЯ, ЧТО
НАШИ ДРУЗЬЯ ИСПОЛЬЗУЮТ
НАДЕЖНЫЕ ПАРОЛИ, ЧТОБЫ ИХ
АККАУНТЫ НЕ ВЗЛОМАЛИ.



Если пароль длиной не менее 8 символов:
показать 😊

Иначе:
показать ☹️

Кажется, мы свое дело сделали. Именно об этом мечтают все учителя, кураторы, наставники, партнеры и члены сообщества «Программирование для девочек»: видеть, как девочки вместе вдохновенно творят, воплощают в жизнь свои мечты и пишут программы, чтобы помочь другим и изменить мир.

Именно этим занималась Кая Томас с тех пор, как в старших классах научилась программировать. Ее пример доказывает, что программирование — это сила, с помощью которой можно совершить много хорошего.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

СОВЕТЫ ПРОФЕССИОНАЛОВ СОЗДАТЕЛЬНИЦА ПРИЛОЖЕНИЯ «МЫ ТОЖЕ ЧИТАЕМ»



Кая Томас с детства любила читать, но ей было сложно представить себя на месте героев большинства книг. В библиотеке и книжном магазине обычно предлагали книги о приключениях белых людей. Поэтому Кая решила создать приложение, которое помогло бы детям разных национальностей находить книги и истории, написанные именно для них. Сейчас она расскажет о том, как воплощала свою амбициозную идею в жизнь.

В старших классах я почти каждый день ходила в библиотеку и брала новые книги. Я стала замечать, что среди тех девочек, о которых я читала, ни у одной не было такой кожи и таких волос, как у меня. Из-за этого я чувствовала себя невидимкой и очень расстраивалась. Почему никто не пишет историй для меня? Почему я не могу их найти? Вот бы где-нибудь был список таких книг, мечтала я. Так и родилась эта идея. Я всегда обожала технологии и активно пользовалась соцсетями, но никогда не думала, что смогу сама создать что-нибудь подобное. Мне это просто не приходило в голову.

А потом в колледже я начала изучать информатику и просто влюбилась. Я поняла, что теперь могу осуществить свою идею.

Но порой приходилось нелегко.

Я решила специализироваться на информатике и очень волновалась. После нескольких вводных курсов мне пришлось учиться по про-

САЙТЫ, МОБИЛЬНЫЕ ПРИЛОЖЕНИЯ И ИНТЕРНЕТ-БЕЗОПАСНОСТЬ

грамме, где было много математических предметов. Математику я знала не слишком хорошо, поэтому я сомневалась, что мне вообще удастся специальность. К счастью, один из профессоров поддержал меня и помог не бросить курс. У меня были не самые высокие оценки, но я прошла, и меня это очень вдохновило. Я поняла, что смогу, нужно только не терять веру в себя.

Вы будете сомневаться, будете хотеть все бросить, но не дайте этим порывам взять верх. Осознайте их, а потом отодвиньте в сторону и уверенно ступайте к своей цели.

Кая всегда напоминает своим ученикам, что они уже обладают нужными знаниями, уверенностью и решимостью. Для нее отдача — очень важная часть работы.

Я хочу поделиться своим опытом и наблюдениями. Хочу сказать: «Вы можете обогнать меня, можете добиться большего успеха, чем я». Если мы хотим возвращаться в технологической индустрии, то наш долг — дать как можно большему числу женщин, цветных, людей из разных слоев общества возможность попасть сюда, чтобы вместе с нами изобретать новые технологии и решать проблемы своих сообществ. Единственный способ сделать это — сказать нашим детям и тем, кто придет после нас: «Мы всегда вас поддержим. Вы можете сделать больше, чем мы, вы можете справиться лучше».

ЗАКЛЮЧЕНИЕ

НУ ВОТ ТЫ И ДОБРАЛАСЬ СЮДА! КНИГА ЗАКАНЧИВАЕТСЯ, НО ДЛЯ ТЕБЯ ВСЕ ТОЛЬКО НАЧИНАЕТСЯ. ЕСЛИ МЫ СПРАВИЛИСЬ СО СВОЕЙ РАБОТОЙ, ТО ТЫ УЖЕ ГОТОВА НАЧИНАТЬ ПРОГРАММИРОВАТЬ. ТЕБЯ ЖДЕТ ЦЕЛЫЙ МИР, ПОЛНЫЙ ТВОРЧЕСТВА, ИЗОБРЕТАТЕЛЬСТВА И ВОДОХНОВЕНИЯ.

ЧТО ТЕПЕРЬ?

Если ты готова начать программировать, первым делом загляни на наш сайт: GirlsWhoCode.com. Мы собрали там множество полезных ссылок, которые помогут тебе сделать следующий шаг в путешествии в мир программирования. Конечно, мы подскажем тебе все, что нужно, чтобы найти или создать клуб «Программирование для девочек» там, где ты живешь. Ты станешь частью сообщества девочек, которые учатся программировать, в своем родном городе, по всей стране и даже в целом мире.

ЗАКЛЮЧЕНИЕ



Послушай, что говорит моя подруга Дебра Стерлинг, успешный инженер и предприниматель. Дебра создала компанию GoldieBlox, выпускающую интерактивные игрушки для девочек, успешно развивает ее и находит время учиться программированию.

ДЕБРА: В колледже я изучала машиностроение, дизайн и взяла пару уроков информатики, а сейчас очень жалею, что не уделила информатике больше внимания. Поэтому недавно я сама начала учиться программировать. Я считаю, что дети должны учиться программировать тогда же, когда они учатся читать и писать. Эти навыки и понимание того, как работает мир, очень важны для того, чтобы строить мир и превращать его в место, где хочется жить.

Все еще не уверена, что программирование — это твое? Что ж, пусть эксперты скажут нам, что они думают по этому поводу.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

Джули и Селеста («Ангел-хранитель»)

ДЖУЛИ: Попробуй. Возьми хотя бы один урок. Нельзя отказываться от того, чего никогда не пробовала, тем более что это открывает та-а-кие возможности! Мне кажется, многие сомневаются, потому что представляют себе программиста как человека, который сидит за ноутбуком по 20 часов в день, но это не так. Программирование — это не просто умение. Для многих оно становится делом всей жизни.

СЕЛЕСТА: И его можно применять где угодно. Чем бы ты ни интересовалась, все будет лучше с программированием.

Трэн («Ярче звук»)

Я знаю, что есть люди, которым стоит только услышать слова «информатика» и «программирование», и они сразу же говорят: «Фу-у-у». Но ты попробуй. Я не думала, что мне понравится программировать, но стоило только начать, как я втянулась. Это перевернуло все мое будущее. Я и дальше хочу работать с компьютерами.

Глори и Захра («Карьера от кутюр»)

ГЛОРИ: Почти все, что ты делаешь, как-то связано с программированием. И его можно комбинировать с чем угодно — модой, расширением возможностей, журналистикой и вообще всем.

ЗАХРА: Я не думала, что стану девочкой, которая умеет программировать. Я не считала себя достаточно умной, чтобы написать простой алгоритм. Мне кажется, большинство девочек не хотят учиться программировать, потому что сомневаются в себе. Вот мой совет: поверьте в себя, потому что вы можете изменить мир своим кодом, строчка за строчкой.

Вот что я хотела сказать.

ТО, ЧТО ТЫ УЧИШЬСЯ ПРОГРАММИРОВАНИЮ, ВАЖНО, ПОТОМУ ЧТО ТЫ ВАЖНА. Важны твой голос, твоя точка зрения, твои идеи

ЗАКЛЮЧЕНИЕ

и творческая энергия. Женщины составляют 50% населения земного шара. Мы — половина населения Земли! Так почему мы не можем быть половиной генеральных директоров или предпринимателей? Почему мы не составляем половину мировых лидеров и политических деятелей, половину изобретателей и тех, кто меняет мир?

На самом деле мы можем такими стать, просто пока у нас не было возможности. Мы сделаем это, освоив технологии, которые так важны для карьер и экономики будущего. Ты храбрая, сильная, умная. Ты — девочка, которая умеет программировать. И ты способна изменить мир.



А ТЕПЕРЬ ПРИШЛО ВРЕМЯ ОСВОИТЬ ЯЗЫКИ ПРОГРАММИРОВАНИЯ!

Поздравляем, ты дочитала эту книгу и наверняка сейчас тебе не терпится приступить к изучению языков программирования и созданию собственных классных проектов. Сделать это будет гораздо проще с наглядными пошаговыми самоучителями для детей от издательства «Эксмо».



ПРОГРАММИРОВАНИЕ ДЛЯ ДЕТЕЙ. ДЕЛАЙ ИГРЫ И УЧИ ЯЗЫК SCRATCH! АВТОР: ЭЛ СВЕЙГАРТ

Популярная во всем мире наглядная платформа Scratch — это то, что нужно, чтобы понять, как работают языки программирования, как строится код, но при этом не заскучать. В этой книге ты найдешь массу понятных примеров, забавные цветные иллюстрации и много крутых игр, которые сможешь создать своими руками!



PYTHON ДЛЯ ДЕТЕЙ И РОДИТЕЛЕЙ. АВТОР: БРАЙСОН ПЭЙН

Хочешь освоить один из самых популярных языков программирования? Тогда твой выбор Python, ведь этот язык не только широко используется, но и прекрасно подходит для новичков. А благодаря этой книге, в которой есть много цветных картинок, понятных объяснений и интересных заданий, ты научишься рисовать графику, шифровать секретные послания, создавать приложения и веселые игры.

НЕСКОЛЬКО ПРОСТЫХ ШАГОВ ДЛЯ РЕАЛИЗАЦИИ СВОЕГО ПРОЕКТА МЕЧТЫ С «КОДАБРА»

1. Если ты очень хочешь создать свою игру, сайт или приложение, но думаешь, что в твоём возрасте это невозможно, то скорее всего ты просто не знаешь, на что способна на самом деле! Итак, первый шаг — поверь в себя и действуй!
2. Сформулируй и запиши идею. А если сделаешь пару простых рисунков на эту тему — будет просто замечательно! Старайся продумать как можно больше деталей.
3. Расскажи об этом друзьям. Покажи им все свои наработки и попытайся заинтересовать их. Работать в команде гораздо легче и интереснее!
4. Распределите роли: если ты любишь рисовать — лучше всего исполнять роль художника, нравится математика — роль программиста точно твоя.
5. Определите очередность выполнения задач! Если вы решили делать, например, игру-приключение, то начинать с проработки синего попугая, которого можно встретить на 15 уровне игры в секретной локации, — плохая идея. Лучше всего начинать с основных моментов, например, подумать и реализовать перемещение главного героя по вашему миру.
6. Когда вы решили, с чего начать, нужно определить задачи для каждого! Тут тоже нет ничего сложного: если ты программист, твоя задача сделать механику передвижения, если художник — займись проработкой персонажа. Очень важно обсудить это всей командой!
7. Теперь, когда вы всё продумали и решили, переходим к самой важной и захватывающей части — реализации. Если вы ничего не знаете — не паникуйте, это нормально, тем более в вашем возрасте. Лучше начать с выбора инструмента. Для создания игры лучше всего подойдет какая-нибудь программа, например Unity или GameMaker. Если вы рисуете, Photoshop — отличный выбор. Попробуйте поискать в Интернете наиболее подходящие для вас программы.
8. Когда инструмент выбран, его нужно изучить! К счастью, сегодня существует множество самоучителей по тем или иным программам и языкам программирования. Загляни на сайт <https://eksmo.ru/professionalnaia-literatura/kompyuternaya-literatura/>, чтобы выбрать самоучитель по нужному языку.
9. Возникли вопросы? Обсуди их со сверстниками в сообществах «Кодабра» Facebook или «ВКонтакте» — www.facebook.com/codabra/ или https://vk.com/codabra_org. С «Кодаброй» свои первые digital-проекты сделали уже более 30 000 детей и подростков.
10. Никогда не опускай руки! Впереди тебя ждет много замысловатых и интересных задач, с которыми ты обязательно справишься!

СЛОВАРЬ ТЕРМИНОВ

АЛГОРИТМ

Алгоритм — это последовательность шагов, которые компьютер выполняет, чтобы справиться с задачей. Можно составить алгоритм для чего угодно: от решения задач по математике до написания музыки! (См. «Алгоритмическая композиция».)

АЛГОРИТМИЧЕСКАЯ КОМПОЗИЦИЯ

Алгоритмическая композиция — это написание музыки на компьютере с помощью алгоритмов или пошаговых инструкций. Представь себе симфонию, целиком написанную роботами!

АППАРАТНОЕ ОБЕСПЕЧЕНИЕ

Аппаратное обеспечение — это механические части компьютера или другого устройства: клавиатура, монитор, карта памяти и т. п.

БИБЛИОТЕКА

Библиотека — это собрание ресурсов, которые ты можешь использовать в своей программе. Одну библиотеку можно использовать для разных программ, чтобы не писать повторно одинаковый код.

БЛОК-СХЕМА

С помощью картинок и стрелок блок-схема показывает, что и в каком порядке будет происходить

в программе. Рисуя блок-схему, ты проходишь логические шаги, которым будет следовать твоя программа.

ВВОД

Информация и указания, которые ты даешь компьютеру, называются вводом. Когда ты печатаешь на клавиатуре или разблокируешь смартфон отпечатком пальца, ты вводишь в компьютер задачу, и он ее выполняет.

ВИЗУАЛИЗАЦИЯ

Визуализация — это создание на компьютере картинок, графиков или анимаций, которые помогают тебе рассказать историю на основе данных. Если ты когда-нибудь видела интерактивную карту или инфографику, то ты видела визуализацию!

ВЫВОД

Вывод — это то, что делает компьютер после получения ввода и выполнения программного кода. Например, когда ты касаешься иконки приложения (ввод), компьютер выполняет код и приложение открывается (вывод)!

ВЫЧИСЛИТЕЛЬНОЕ МЫШЛЕНИЕ

Вычислительное мышление помогает нам логически решать сложные задачи: мы разбиваем их на меньшие части, ищем закономерности и с помощью этой информации находим поэтапное решение.

СЛОВАРЬ ТЕРМИНОВ

ДАННЫЕ

Данные — это информация, которую ты вводишь в компьютер, чтобы он выполнил задачу или произвел вычисление.

ДВОИЧНОЕ ЧИСЛО

Двоичное число — это обычное число, выраженное в двоичной системе, то есть записанное единицами и нулями. Каждый разряд такого числа называется битом.

ДВОИЧНЫЙ КОД

Двоичный код превращает слова или указания для компьютерного процессора в серии единиц и нулей, которые поясняют компьютеру, что делать. Например, слово «привет» в двоичном коде будет выглядеть как 1101000010111111101000110000000110100001011100011010000101100101101000010110101110100110000010.

ДИЗАЙН

Дизайн — это план или схема, которые поясняют, как что-то должно выглядеть и работать. Например, дизайн платья поясняет, как сшить это платье, а дизайн приложения для iPhone показывает, как это приложение должно работать. Люди, которые создают такие планы и схемы, называются дизайнерами.

ДОКУМЕНТАЦИЯ

Документация — это описание сайта или приложения, составленное для пользователей. Сейчас документацию в основном выкладывают в онлайн, но бывают и бумажные руководства пользователя или владельца.

ЖАНР

Игровые жанры — это группы компьютерных игр, которые ставят перед

игроком схожие задачи. Например, образовательные игры — это один жанр, а квесты — уже другой.

ИНФОРМАТИКА

Информатика — это наука о компьютерах и о том, как их можно использовать. Специалисты по информатике пишут программы, которые делают самые разные вещи: помогают лечить людей, сочиняют музыку и даже рисуют картины.

КИБЕРБЕЗОПАСНОСТЬ

Кибербезопасность — это защита информации на твоём компьютере, телефоне и других устройствах, чтобы её никто не повредил и не украл.

КОД

С помощью кода мы описываем шаги, которые должна сделать компьютерная программа, то есть пишем инструкцию для компьютера.

КОМПИЛЯТОР

Когда ты пишешь программу, то используешь английские или русские слова, чтобы пояснить компьютеру, что делать. Компилятор переводит код, написанный человеком, на «машинный язык», понятный компьютеру.

КОМПЬЮТЕРНОЕ ПРОГРАММИРОВАНИЕ

Когда ты пишешь программу или программируешь компьютер, ты указываешь ему, что делать. Компьютеры кажутся умными, но на самом деле они ничего не могут без программ, написанных людьми!

ЛОГИКА

Упорядоченный ход мыслей, понятный компьютеру.

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

ЛОГИЧЕСКАЯ ОШИБКА

Логическая ошибка — это баг в коде программы, который не дает программе выполняться правильно. Например, условная конструкция оказывается ложной, когда должна была быть истинной.

ЛОГИЧЕСКОЕ ЗНАЧЕНИЕ

Логическое, или булево (по имени английского математика Джорджа Буля), значение — это одно из двух значений: «истинно» (на англ. true) или «ложно» (на англ. false). Логические значения используются в условных конструкциях. Если утверждение истинно, то компьютер выполняет одно действие; а если утверждение ложно, компьютер выполняет другую операцию.

МАКЕТ

Макет — это набросок того, как будет выглядеть и работать сайт или приложение. Макет состоит из обычных ячеек и линий. Он похож на пазл, который показывает, куда что встанет и как все это будет выглядеть вместе.

МОЗГОВОЙ ШТУРМ

В процессе мозгового штурма вы все вместе придумываете, как решить какую-нибудь важную проблему. Для мозгового штурма плохих идей нет!

«НЕ ПОВТОРЯЙСЯ»

Принцип «Не повторяйся» (или DRY, от англ. Don't Repeat Yourself) означает, что не нужно писать одинаковые кусочки кода снова и снова, иначе потом его трудно будет читать и исправлять. Если надо сделать что-то несколько раз, можно написать функцию или воспользоваться циклом.

ОТЛАДКА

Редко программа работает идеально с первого раза. Отладить

программу — значит найти, что в программе пошло не так, и исправить эту ошибку, или выловить баг (от англ. bug — «жук»). Название «баг» произошло от случая, когда Грейс Хоппер обнаружила в компьютере мотылька (настоящее насекомое!), из-за которого компьютер и отказывался работать.

ПАРАМЕТР

Параметр — это вид переменной, который применяется в функции. Например, в функции `def f(x): ...` `x` — это параметр. Иногда слова «параметр» и «аргумент» означают одно и то же.

ПЕРЕМЕННАЯ

Переменные — это что-то вроде коробок, которые используются в программе для того, чтобы хранить и запоминать информацию. Переменная может содержать число, строку букв и даже информацию о том, истинно что-то или ложно!

ПОЛЗУЧИЙ УЛУЧШИЗМ

Ползучий улучшизм — стремление постоянно добавлять в проект что-то новое, что пользователю, может быть, вовсе и не надо. Эти новые свойства «вползают» внутрь проекта, отчего он разбухает и становится более сложным, чем нужно.

ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС

Пользовательский интерфейс — это способ для взаимодействия человека с любым устройством: от мобильного приложения до микроволновки. Если интерфейс хорошо продуман, то с устройством можно свободно работать без всякого обучения. Именно поэтому даже малыши могут пользоваться планшетами!

СЛОВАРЬ ТЕРМИНОВ

ПРИЛОЖЕНИЕ

Приложение — это программа, которая может работать на компьютере, в Интернете или на небольшом устройстве, таком как смартфон или планшет. Веб-приложения (например, сайты социальных сетей или календари) отличаются от сайтов. Сайт сразу предоставляет информацию, а для работы веб-приложения нужно ввести данные. Приложения могут быть очень разными: для работы с текстом, для игры, обработки фотографий и общения с друзьями в соцсетях.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Программное обеспечение (ПО) — это все, что может храниться в электронном виде, например программы или инструкции. ПО делится на две категории: системное и прикладное. Системное ПО — это операционная система, на которой работает твой компьютер. Прикладное ПО — это программы, с которыми работаешь ты (такие как текстовый редактор или программа для обработки фотографий).

ПРОГРАММНЫЙ ИНТЕРФЕЙС ПРИЛОЖЕНИЯ

Программный интерфейс приложения (англ. application programming interface, API) — это система правил, которая позволяет приложениям взаимодействовать друг с другом. Например, вместо того, чтобы разрабатывать свою собственную карту, можно воспользоваться программным интерфейсом карт Google и разместить их карту у себя на сайте или в приложении.

ПРОЦЕДУРАЛЬНОЕ ИСКУССТВО

Процедуральное искусство — это искусство, в создании которого (хотя бы частично) участвуют компьютеры.

Например, чтобы анимировать волосы персонажа, можно написать отдельный код для каждого волоска или с помощью кода реалистично размножить волоски и сэкономить время. Аниматоры Pixar использовали процедуральное искусство, чтобы волосы принцессы Мериды из «Храброй сердцем» выглядели естественно и вели себя как копна непослушных кудряшек. Они моделировали движение локонов, вместо того чтобы анимировать каждый волосок по отдельности.

ПРОЦЕСС

Процесс — это программа, запущенная на твоём компьютере или устройстве. Одно приложение может запускать несколько разных процессов одновременно!

ПСЕВДОКОД

Псевдокод — это описание программы, которое могут прочесть люди, но не компьютеры. Очень удобно расписать логику программы на псевдокоде, а затем перевести ее в код на языке программирования.

РАЗРАБОТКА КОМПЬЮТЕРНЫХ ИГР

Разработка — это процесс задумки и создания компьютерной игры. Участники «Программирования для девочек» делают замечательные игры, которые затрагивают очень важные темы, например вымирание животных или социальные проблемы.

РАЗРАБОТКА – СОЗДАНИЕ – ТЕСТИРОВАНИЕ

В ходе цикла «разработка — создание — тестирование» ты разрабатываешь что-то, затем создаешь, то есть воплощаешь в жизнь, и тестируешь. После этого можно использовать полученные

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

знания, чтобы усовершенствовать разработку. Это цикл, и он повторяется до тех пор, пока ты не останешься довольна результатом.

РАСКАДРОВКА

Раскадровка — это комикс, с помощью которого можно спланировать, что будет происходить в твоей игре или программе. Бывают очень красивые раскадровки, но на самом деле совершенно неважно, как они выглядят. Чтобы понять, как будет развиваться действие, достаточно рисунка в стиле «палка, палка, огуречик».

РОБОТОТЕХНИКА

Робототехника — это область информатики, которая занимается созданием машин, использующих программы. Запрограммированные машины, то есть роботы, могут выполнять различные задачи и способны на поразительные вещи: от помощи докторам до исследования океанских глубин.

САЙТ

Сайт — это место в Интернете, которое дает информацию, но не требует обратной связи. Сайты чем-то похожи на школьные собрания: приходит какой-нибудь человек и рассказывает вам о чем-нибудь, но вы сами ничего ему не говорите. Немного освоив HTML и CSS, можно сделать отличный сайт!

СИНТАКСИС

Синтаксис — это порядок букв, чисел и символов в коде (как и в предложениях на русском). От правильного синтаксиса зависит работоспособность программы. Сравните фразы «Привет тобой! Познакомиться, приятно

с очень» и «Привет, очень приятно с тобой познакомиться!». Слова одинаковые, вся разница в синтаксисе.

СИНТАКСИЧЕСКАЯ ОШИБКА

Синтаксическая ошибка происходит, когда в синтаксисе кода что-то не так: лишний пробел, не хватает скобки или потерялась точка с запятой. В компьютерной программе очень важна точность.

СОТРУДНИЧЕСТВО

Когда двое или больше людей работают вместе над проектом или задачей, они сотрудничают. Благодаря сервисам онлайн-сотрудничества, таким как GitHub, программисты из разных регионов могут одновременно работать над одной и той же программой.

СТРОКА

Строка — это разновидность данных, состоящая из символов: цифр, букв и/или пробелов. Отличить строку легко, она всегда заключается в кавычки. Вот примеры строк: "программирование для девочек", или "программирование для девочек", или "12345".

ТЕКСТОВЫЙ РЕДАКТОР

Это приложение, в котором можно набирать текст, в том числе программный код. Существует множество текстовых редакторов, предназначенных специально для программистов. В таких приложениях есть комбинации клавиш для быстрого вызова и автоматического распознавания ошибок.

УСЛОВНАЯ КОНСТРУКЦИЯ

Условная конструкция — это элемент кода, который выполняется только при соблюдении определенного условия.

СЛОВАРЬ ТЕРМИНОВ

Поэтому такая конструкция и называется условной.

ФУНКЦИЯ

В программе функция — это последовательность шагов, связанных воедино, как задача по математике. Когда ты вводишь в функцию параметры, она обрабатывает их и выводит результат.

ЦИКЛ

Цикл — это способ написать один элемент кода так, чтобы он повторился несколько раз. Если я хочу нарисовать квадрат, я могу написать цикл «прямо, потом направо» и повторить его четыре раза вместо того, чтобы писать восемь строчек кода.

ЧЕТЫРЕ ЯДРА

Четыре ядра информатики (англ. Core4 computer science concepts) — это термин, изобретенный участницами «Программирования для девочек». «Четыре ядра» означают переменные, циклы, условные конструкции и функции. Мы считаем их четырьмя основными принципами информатики, потому что они присутствуют практически во всех языках программирования.

ЭФФЕКТИВНОСТЬ

Эффективность компьютерной программы сравнивает результат работы программы с вложенными в нее усилиями. Самые эффективные программы быстро обрабатывают данные и не занимают много компьютерной памяти — как гепард, который бежит очень быстро и тратит мало энергии.

ЯЗЫК ПРОГРАММИРОВАНИЯ

Язык программирования — это набор правил и инструкций, которым пользуются, чтобы написать компьютерную программу. Существует множество разных языков программирования, которые годятся для различных целей.

HELLO, WORLD!

Hello, World! (русск. Привет, мир!) — простенькая программа, которая выводит на экран строку с приветствием. Первая программа, которую пишут начинающие программисты.

NASA

Национальное управление по аэронавтике и исследованию космического пространства, США. Занимается научными исследованиями воздушного и космического пространства.

Благодарности

За эти пять лет я видела, как десятки тысяч девочек из всех американских городов и штатов учатся программировать и создавать вещи, которые решают наши самые актуальные проблемы, такие как насилие с применением огнестрельного оружия, изменение климата, рак, бедность. Эта книга посвящается им и всем тем, кого они вдохновили, как и меня. Это призыв к девочкам всей страны, всего мира: учитесь программировать и присоединяйтесь к нам, чтобы внести свой вклад.

Я хотела бы поблагодарить Сару Хатт за то, что она перевела сложный мир программирования на понятный всем нам язык. Ты стала одной из нас, и мы так легко тебя не отпустим.

Спасибо Джеффу Стерну за его неопенимую работу над книгой. Я пришла к нему с этим проектом около трех лет тому назад, и он сразу согласился. Я знала, что Джефф, великолепный учитель и творческий человек, сможет превратить изучение базовых принципов информатики в интересное и веселое занятие.

Спасибо всей команде «Программирования для девочек», особенно тем, кто помогал работать над этим проектом: нашей главной программистке Эмили Рейд, а также Деборе Сингер, Джессамине Бартли-Мэттьюс, Клэр Кук, Крисси Зиккарелли, Лее Гиллиам, Саре Джудд, Ханне Галли, Худе Куреши, Шарлотте Стоун, Эллен Маккаллах и Эрику Гюнтеру.

Спасибо Ричарду Пайну, моему потрясающему агенту, у которого на эту книгу

были грандиозные планы — грандиознее, чем я могла себе представить. Для меня было честью работать с его замечательной командой из Inkwell, в том числе с Элизой Ротштейн и Натаниэлем Джексом.

Спасибо иллюстратору Андреа Цурими, чьи выразительные рисунки оживили наших персонажей и их истории.

Спасибо Кендре Левин, королеве издателей, за ее познания и руководство. Ты сделала эту книгу реальностью. Также я хотела бы поблагодарить Кейт Реннер, нашего арт-директора, и Бетани Брайан, нашего редактора.

Я благодарю всю команду Viking и особенно Кена Райта. Мы с удовольствием включились в сотрудничество с маркетинговой группой и книготорговой сетью Penguin Young Readers Group, чтобы донести эту книгу до девочек по всему миру.

Спасибо всем прекрасным женщинам, которые поделились своими вдохновляющими историями: Доне Бейли, Аяне Говард, Даниэль Фейнберг, Дебре Стерлинг, Кае Томас и Челси Хау.

Спасибо Анни Х., Глори К., Джули П., Захре Л., Идализ Д., Кенише Дж., Марии М., Нани Н., Селесте Б., Серене В., Эмбер С., Фейт У., Эмили Д., Янаншали С. и Ясмин Л. за то, что разрешили описать в книге их проекты.

Миллион благодарностей всем ученицам «Программирования для девочек» и всем тем, кто учится программировать, за интересные истории и отзывы, сформировавшие книгу, которую ты сейчас держишь в руках.

Алфавитный указатель

- Android-устройства, 148
- Apple-устройства, 148
- Arduino (язык программирования), 95
- C (язык программирования), 95
- C# (язык программирования), 93, 95
- Hershey's (производитель шоколада), 128
- HTML/CSS (языки разметки), 94, 115
- Java (язык программирования), 93, 95, 148
- JavaScript (язык программирования), 94
 - в «Зеркале погоды» (приложение), 58–59
 - и «Карьера от кутюр» (приложение), 115
 - и отладка ошибок, 101
 - примеры, 92, 100–101
- LEGO Mindstorms Nxt (язык программирования), 94
- Maya (встроенный язык программирования), 95
- MIT App Inventor (визуальный язык программирования), 153
- Obj=C (язык программирования), 148
- Pixar, 132–133
- Processing (язык программирования), 95
- Python (язык программирования), 92, 94
- Ruby (язык программирования), 92
- SuperBetter (игра), 121
- Swift (язык программирования), 92, 95, 148
- Алгоритмическая композиция, 129
- Алгоритмы, 45–50
 - библиотеки кода для, 55–56
 - в псевдокоде, 50
 - и блок-схема, 84
 - определение, 46
 - примеры, 48–49
 - функции, 46–48
 - циклы в, 51–52
- «Ангел-хранитель» (приложение), 151–153
- Антонелли, Кэтлин (Кей) Макналти Мокли, 27
- «Аполлон» (программа полетов на Луну), 28–30
- Аппаратное обеспечение, 19
- Бартик, Джин Дженнингс, 27
- Безопасность и конфиденциальность в Интернете, 149–151
- Безопасность онлайн, 150–153
- Библиотеки кода, 55–56
- Биты, 89
- Блок-схема, 78–85
- Бэббидж, Чарльз, 25, 26
- Ввод, 21–22, 37
- Вдохновение
 - важность быть открытым для, 133
 - доска вдохновения, 66
- Веб-приложения для мобильных устройств, 148

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

- Визуализация данных, 127
 Возможности, важность быть открытым, 133
 Возможности для девочек изучать программирование, 158–159
 Встроенные приложения, 148
 Вывод, 21–22, 37
 Высадка НАСА на Луне, 28–29, 30
 Вычислительное мышление, 35–37
- Гамильтон, Маргарет, 30
- Данные, 39
 шифрование, 149, 150
 Двоичный код, 89–90, 91
 Деление на ноль (ошибка), 101
 Дизайнеры, 72
 Дроны, 137
 Друзья
 и вопросы по отладке, 104
 сотрудничество с, 67
- Женщины, их достижения и место в мире, 161
- «Зеркало погоды» (приложение), 57–59
- Иллюстрации, 115
- «Иначе» в условных конструкциях, 41–42, 49
- Интегрированная среда разработки (ИСР), 107
- Интернет
 безопасность и конфиденциальность в 149–155
 заря, 30
 и личная безопасность онлайн, 150–153
- и отладка ошибок, 103
 и разработка сайтов, 147
 разработка интернет-приложений, 147, 148
 советы профессионального разработчика приложений, 156–157
- Картины анимированные, 126
 «Карьера от кутюр» (приложение), 114–116
- Карьеры в компьютерной индустрии, 9
 Карьеры в точных науках, 10–11
 Кибербезопасность, 149–150
 Кобол (язык программирования), 91
 Комиксы анимированные, 126
 Компиляторы, 91–92
 Компьютерные игры, 111–121
 жанры, 112–113
 «Карьера от кутюр» (приложение), 114–116
 поиск идей для, 117–119
 советы профессионального разработчика, 120–121
- Компьютеры
 аппаратное и программное обеспечение, 19
 буквальное мышление, 34, 47
 и ввод — обработка — вывод, 21–22, 37
 и вычислительное мышление, 35–37
 и двоичный код, 89–90
 и расцвет персональных компьютеров, 28–29
 первая программа для, 26
 первые компьютеры, 24–25
 эволюция, 23–25

АЛФАВИТНЫЙ УКАЗАТЕЛЬ

- Конфиденциальность, 149–151
 Космические миссии
 миссия по исследованию Марса,
 142
 полет на Луну, 28–29, 30

 Лавлейс, Августа (Ада), 26
 Логика, 38
 и псевдокод, 86
 логические ошибки, 100
 Логические значения, 39

 Макеты, 73
 Марс, миссия по исследованию, 142
 Массачусетский технологический
 институт, 28, 30
 Медицина и роботехника, 136, 143
 Мельцер, Мэрлин Уэскофф, 27
 Мода и дизайн, 128
 Мозговой штурм, 64–67
 в проектах по цифровому
 искусству и дизайну,
 130–131
 в роботехнике, 139–141
 Мокли, Джон, 26
 «Мы тоже читаем» (приложение),
 156–157

 Нанороботы, 136
 НАСА, 28–29, 30, 142
 «Не повторяйся» (принцип), 51

 Обработка данных, 21–22, 37
 Отладка ошибок, 99–109
 и принятие собственного
 несовершенства, 108–109
 и типы ошибок, 100–101
 нахождение ошибки, 102
 перерывы в, 107–108
 помощь с, 102–105
 проговаривание вслух, 105–106
 путем вычитывания кода, 106
 с помощью интегрированной
 среды разработки, 107
 с помощью функции вывода,
 106–107
 Ошибки: См. Отладка ошибок

 Пайка, 125
 Параметры функций, 52–53
 Переменные, 38–39, 42
 в алгоритмах, 49
 в блок-схемах, 82–83, 85
 в псевдокоде, 86
 в разработке компьютерных игр,
 119
 Пользовательский интерфейс, 72,
 147
 Потеря веры в себя и как с ней
 справиться, 157
 Программирование (определение),
 18
 «Программирование для девочек»
 клубы и программы, 12, 158
 основательница, 8
 сайт, 158
 Программная инженерия, 30
 Программное обеспечение, 19
 Программный интерфейс
 приложения, 57
 Процедуральное искусство, 126
 Псевдокод
 в проектах по цифровому
 искусству и дизайну, 131
 и «Карьера от кутюр»
 (приложение), 115
 и «Не повторяйся» (принцип), 51
 функции, 50, 86–87

ПРОГРАММИРУЙ! САМОУЧИТЕЛЬ ДЛЯ ДЕВОЧЕК

- «Равно» (знак) в условных конструкциях и связанные с ним ошибки, 101
- Развитие технологий, 23–24
- Разработка, 68–75
 - визуализация продукта, 72–74
 - выяснение, будет ли спрос на продукт, 70
 - выяснение, нужен ли такой продукт, 68–69
 - и кибербезопасность, 154–155
 - и ползучий улучшизм, 74–75
 - и пользовательский интерфейс, 72, 147
 - исследования аналогичных продуктов, 69
 - с учетом наличия времени, 71
- Разработка сайтов, 147
- Разработка интернет-приложений, 147–148
- Разработка — создание — тестирование, 63–64
- Раскадровка, 73–74
- Резервная копия, 96
- Решение задач как суть программирования, 18, 88, 105
- Робот-поводырь (проект), 138–141
- Роботы, 135–143
 - дроны, 137
 - мозговой штурм, 139–141
 - нанороботы, 136–137
 - робот-поводырь (проект), 138–141
 - советы профессионального роботехника, 142–143
 - типы, 136–137
- Светодиоды
 - платья со светодиодами, 128
 - система «Ярче звук», 124–126
- Синтаксис, 86
- Синтаксические ошибки, 100–101
- Система названий файлов, 96
- Скретч (язык визуального программирования), 94
- Сотрудничество, 67, 139, 153
- Спенс, Фрэнсис Билас, 27
- Старшие товарищи, 104
- Стерлинг, Дебра, 159
- Строки, 39
- Тейтельбаум, Рут Лихтерман, 27
- Текстовые редакторы, 107
- Техническое обслуживание сетей, 149
- Томас, Кая, 155, 156–157
- Трехмерная печать, 128
- Условные конструкции, 41–42
 - в алгоритмах, 49
 - в блок-схемах, 85
 - в псевдокоде, 86
 - в разработке компьютерных игр, 119
- Учительская помощь, 104
- Фейнберг, Даниэль, 131, 132–133
- Функции, 42, 52–54, 56
- Функция вывода, 106–107
- Хауи, Челси, 119, 120–121
- Ховард, Аянна, 141, 142–143
- Холбертон, Фрэнсис Элизабет (Бетти), 27
- Хоппер, Грейс Мюррей, 91, 92
- Художники, 126
- Циклы, 40–41
 - бесконечный цикл (ошибка), 101
 - в блок-схемах, 85
 - в псевдокоде, 86

АЛФАВИТНЫЙ УКАЗАТЕЛЬ

- в проектах по цифровому искусству и дизайну, 131
- в разработке компьютерных игр, 119
- функции, 49, 51–52
- Цифровое искусство и дизайн, 123–133
- в визуализации данных, 127
- в изобразительном искусстве, 126–127
- в моде и дизайне, 128
- в музыке, 128–129
- и система «Ярче звук», 124–126
- мозговой штурм, 130–131
- советы профессионального оператора-постановщика, 132–133
- Четыре ядра информатики, 42–43, 52
- Эволюция технологий и ее скорость, 23–24
- Эккерт, Дж. Преспер, 26
- Электронный числовой интегратор и вычислитель (ЭНИАК), 26, 27
- Языки программирования
 - выбор, 88, 92–95
 - и вопросы по отладке, 103
 - и двоичный код, 91
 - изучение, 31
 - компиляторы, 91–92
 - правила, 37–38

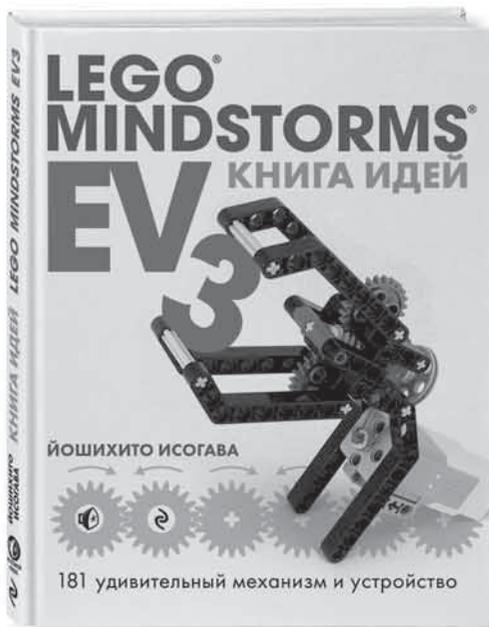
САМОЕ ПОДРОБНОЕ РУКОВОДСТВО ДЛЯ НАЧИНАЮЩИХ ПО ПОСТРОЙКЕ И ПРОГРАММИРОВАНИЮ РОБОТОВ!



Все, что нужно знать, чтобы собирать и программировать роботов – от самых простых до очень сложных.

Для каждой модели дается список нужных деталей, минимальное объяснение и много цветных фотографий под разными углами, чтобы вы смогли собрать ее без пошагового объяснения.

На основе этих книг вы сможете создавать и программировать своих уникальных роботов.



КОГДА ВЫ ДАРИТЕ КНИГУ, ВЫ ДАРИТЕ ЦЕЛЫЙ МИР

ХОТИТЕ ЗНАТЬ БОЛЬШЕ?

Заходите на сайт:

<https://eksmo.ru/b2b/>

Звоните по телефону:

+7 495 411-68-59, доб. 2261



ВАШ ЛОГОТИП
НА ОБЛОЖКЕ

ВАШ ЛОГОТИП НА КОРЕШКЕ

ОБРАЩЕНИЕ
К КЛИЕНТАМ
НА ОБЛОЖКЕ

УЧИТЬСЯ ПРОГРАММИРОВАТЬ И МЕНЯЙ МИР К ЛУЧШЕМУ!

ПРОГРАММИРОВАНИЕ - ЭТО ВЕСЕЛО И ПРОСТО!

ХОЧЕШЬ СОЗДАТЬ СОБСТВЕННУЮ ИГРУ, ФИЛЬТР ДЛЯ ФОТОГРАФИЙ ИЛИ ПРИЛОЖЕНИЕ, КОТОРОЕ ПОМОГАЕТ ВЫБРАТЬ ПРИЧЕСКУ В ЗАВИСИМОСТИ ОТ ПОГОДЫ? ИЛИ ТЫ СОБИРАЕШЬСЯ РАЗРАБОТАТЬ ПРОГРАММУ, КОТОРАЯ СДЕЛАЕТ ЭТОТ МИР ЧУТЬ-ЧУТЬ ЛУЧШЕ? А МОЖЕТ БЫТЬ, ТЫ МЕЧТАЕШЬ ОТПРАВЛЯТЬ КОСМИЧЕСКИЕ КОРАБЛИ НА ОРБИТУ? О ЧЕМ БЫ ТЫ НИ МЕЧТАЛА, С ЭТОЙ КНИГОЙ У ТЕБЯ ВСЕ ПОЛУЧИТСЯ!

ЧТО ВНУТРИ:

- ОСНОВЫ ПРОГРАММИРОВАНИЯ
- МНОЖЕСТВО ИЛЛЮСТРАЦИЙ
- ПОНЯТНЫЕ ОБЪЯСНЕНИЯ
- ИНТЕРЕСНЫЕ ЗАДАНИЯ
- ВООДУШЕВЛЯЮЩИЕ ИСТОРИИ

«Я НЕ УМЕЮ ПРОГРАММИРОВАТЬ, НО ЭТА КНИГА **ВДОХНОВИЛА** МЕНЯ ПОПРОБОВАТЬ. Я СОВЕТУЮ ВСЕМ ДЕВЧОНКАМ ПРОЧИТАТЬ ЕЕ И РЕШИТЬСЯ **ИЗМЕНИТЬ МИР**, ИСПОЛЬЗУЯ ТЕХНОЛОГИИ».

МАЛАЛА ЮСУФЗАЙ,
ПРАВООЩИТНИЦА, ЛАУРЕАТ НОБЕЛЕВСКОЙ ПРЕМИИ МИРА

«ПОДЕЛЮСЬ С ВАМИ СЕКРЕТОМ: КАЖДАЯ ЮНАЯ ДЕВУШКА МОЖЕТ **НАУЧИТЬСЯ ПРОГРАММИРОВАТЬ**. ЧТОБЫ НАЧАТЬ, НЕ НУЖНО РАЗБИРАТЬСЯ В МАТЕМАТИКЕ ИЛИ ЗНАТЬ, КАК УСТРОЕН КОМПЬЮТЕР. ДОСТАТОЧНО УМЕТЬ ГОТОВИТЬ ИЛИ ПЛЕСТИ БИСЕРОМ, НАПРИМЕР. ЭТА КНИГА ПОМОЖЕТ КАЖДОЙ ЧИТАТЕЛЬНИЦЕ **НАБРАТЬСЯ ХРАБРОСТИ** И СДЕЛАТЬ СВОИ ПЕРВЫЕ ШАГИ В IT».

ДАРЬЯ АБРАМОВА,
ОСНОВАТЕЛЬ ШКОЛЫ ЦИФРОВОГО ТВОРЧЕСТВА «КОДАБРА»

БОМБОРА

ISBN 978-5-04-093243-6



9 785040 932436 >

«ЭТА КНИГА **ВДОХНОВЛЯЕТ** ЮНЫХ ЖЕНЩИН **СТАТЬ НОВЫМ ПОКОЛЕНИЕМ** ЛИДЕРОВ В ТЕХНОЛОГИЧЕСКИХ ОБЛАСТЯХ».

ДЖЕК ДОРСИ,
СООСНОВАТЕЛЬ И ГЕНЕРАЛЬНЫЙ ДИРЕКТОР
SQUARE И TWITTER