

ДОГОВОРИСЬ С ИИ



ПЕРВАЯ КНИГА О
CLAUDE CODE

ДЛЯ ТЕХ, КТО НЕ ПРОГРАММИРУЕТ

Даниил Глушко

**Договорись с ИИ. Первая
книга о Claude Code для
тех, кто не программирует**

«Автор»

2026

Глушко Д. А.

Договорись с ИИ. Первая книга о Claude Code для тех, кто не программирует / Д. А. Глушко — «Автор», 2026

Вы никогда не программировали — и это нормально. «Договорись с ИИ» написана именно для вас. Ещё пару лет назад создание сайта или программы требовало месяцев изучения языков программирования. Сегодня достаточно уметь внятно объяснить, чего вы хотите, — а всё остальное сделает искусственный интеллект. Эта книга шаг за шагом проведёт вас от первого «Что вообще такое ИИ?» до момента, когда вы своими руками (точнее, словами) создадите работающий сайт и небольшое приложение с помощью Claude Code — инструмента, который пишет код по вашим текстовым командам. Никакого профессионального жаргона без объяснений. Никаких предположений о том, что вы «и так знаете». Только ясный пошаговый путь: установка программы, первый диалог с ИИ, искусство формулировать задачи так, чтобы получать именно то, что нужно, исправление ошибок, работа с базами данных — и в конце книги честный разговор о том, чему в ответах ИИ можно доверять, а что стоит перепроверять.

© Глушко Д. А., 2026

© Автор, 2026

Содержание

Предисловие	5
Как читать эту книгу	6
Часть 1. Первые шаги в мир ИИ	7
Глава 1. Что такое искусственный интеллект простыми словами	7
С чего всё началось	7
От распознавания картинок до разговора	7
Значит ли это, что ИИ «думает»?	7
Зачем ИИ вообще учат писать код	8
Зачем тогда вообще эта книга, если ИИ и так всё сделает?	8
Небольшое резюме главы	9
Глава 2. Что такое Claude и чем он отличается от других ИИ	10
Кто такая компания Anthropic	10
Чем Claude отличается от других языковых моделей	10
Claude как чат и Claude как инструмент внутри программ	11
Что значит «Claude Code» дословно	11
А это точно не опасно — доверять программе доступ к моим файлам?	11
Небольшое резюме главы	12
Глава 3. Знакомство с Claude Code: зачем нужен ИИ-помощник, который пишет код	13
Что вообще значит «программировать»	13
Что меняет Claude Code	13
Из чего складывается «программа» на практике	13
Что Claude Code умеет делать конкретно	14
Чего Claude Code не умеет и не должен делать	14
Кому и зачем это может пригодиться	15
Небольшое резюме главы	15
Часть 2. Установка и настройка	16
Глава 4. Что такое терминал и командная строка	16
Почему нельзя просто скачать программу с сайта и открыть двойным кликом	16
Что такое терминал	16
Где найти терминал на вашем компьютере	16
Первое знакомство: не бойтесь чёрного окна	17
Небольшой словарь для начала работы	18
Опечатки — это нормально	18
Небольшое резюме главы	18
Конец ознакомительного фрагмента.	19

Даниил Глушко

Договорись с ИИ. Первая книга о Claude Code для тех, кто не программирует

Предисловие

Ты держишь (или, скорее, читаешь на экране) книгу, которая должна была появиться намного раньше. Дело в том, что искусственный интеллект перестал быть темой для узкого круга программистов и учёных ещё несколько лет назад, а инструменты вроде Claude Code — программы, которые пишут, редактируют и чинят код вместе с человеком в диалоге, — за последний год превратились из экзотики в то, чем пользуются миллионы людей: от профессиональных разработчиков до маркетологов, дизайнеров, студентов и просто любопытных людей, которые хотят один раз в жизни сделать свой собственный сайт, telegram-бота или программу для расчёта семейного бюджета.

Проблема в том, что почти вся документация и обучающие материалы по таким инструментам написаны программистами для программистов. Открываешь официальный сайт — и первая же строчка звучит как «установите Node.js версии не ниже 18 и выполните команду `npm install -g`». Для человека, который никогда не открывал терминал, это звучит как заклинание на латыни.

Эта книга написана для того человека, который открывает компьютер, чтобы написать письмо, посмотреть фильм или полистать соцсети, и никогда — чтобы «запустить скрипт». Мы пройдем весь путь: от объяснения, что вообще такое искусственный интеллект и почему он вдруг научился писать код, до момента, когда вы своими руками, вместе с Claude Code, соберёте несколько настоящих небольших проектов.

Я буду избегать профессионального жаргона там, где это возможно, а там, где термин необходим — а в программировании без терминов, увы, никуда — я буду его подробно объяснять при первом появлении и добавлю в конец книги словарь, куда можно будет заглянуть в любой момент.

Обещаю: вы не научитесь этому “за один вечер”, как любят писать на обложках книг для чайников. Но если вы прочитаете эту книгу вдумчиво и, что важнее, будете параллельно пробовать всё своими руками, то через несколько недель регулярной практики по вечерам вы будете чувствовать себя с Claude Code так же уверенно, как с обычным текстовым редактором. А может быть, даже увереннее.

Договорились? Тогда переходим к первой главе.

Как читать эту книгу

Прежде чем нырять в содержание, пара слов о структуре.

Часть 1 знакомит вас с понятиями искусственного интеллекта и Claude Code на человеческом языке, без единой команды в терминале. Если вы уже понимаете, что такое ИИ и чем Claude Code отличается от ChatGPT, можно пролистать её быстро.

Часть 2 — это установка программы на ваш компьютер. Здесь впервые появится терминал, и мы разберём его максимально подробно, шаг за шагом, с картинками команд и объяснением каждой буквы.

Часть 3 закладывает необходимый минимум знаний о программировании — не для того, чтобы вы стали программистом, а для того, чтобы понимать, о чём вообще идёт речь, когда Claude Code говорит вам «я создал файл `index.html`» или «нашёл ошибку в функции».

Часть 4 — это практика. Мы вместе, шаг за шагом, создадим несколько настоящих проектов: сайт-визитку, небольшое приложение, скрипт для автоматизации рутины.

Часть 5 идёт немного глубже — туда, куда новичок обычно попадает не сразу, но куда стоит заглянуть, чтобы понимать общую картину: базы данных, API, автоматизация.

Часть 6 — о безопасности, здравом смысле и типичных граблях новичков.

Часть 7 — куда двигаться дальше, когда книга закончится, а желание разбираться — нет. В каждой главе вы встретите три типа вставок:

СОВЕТ: Практический лайфхак, который сэкономит вам время.

ВНИМАНИЕ: Момент, на котором новички чаще всего спотыкаются.

ТЕРМИН: Объяснение нового слова простыми словами.

Заведите отдельный текстовый файл или тетрадь — записывайте туда команды, которые сработали у вас, и вопросы, которые появляются по ходу чтения. Программирование — и работа с ИИ в частности — это навык, который встраивается в руки через повторение, а не через одно только чтение.

Всё готово. Поехали.

Часть 1. Первые шаги в мир ИИ

Глава 1. Что такое искусственный интеллект простыми словами

С чего всё началось

Представьте, что вы учите ребёнка отличать кошку от собаки. Вы не объясняете ему формулу «если уши треугольные и хвост пушистый — это кошка». Вы просто показываете ему сто картинок кошек и сто картинок собак, и в какой-то момент он сам начинает безошибочно их различать, даже не умея сформулировать, по каким именно признакам.

Именно так — очень грубо говоря — работает современный искусственный интеллект. Вместо того чтобы программист вручную прописывал тысячи правил («если пиксель такого-то цвета находится в таком-то месте, то это...»), компьютеру показывают огромное количество примеров, и он сам находит закономерности. Этот процесс называется **машинным обучением**.

ТЕРМИН: Машинное обучение — способ создания программ, при котором компьютер не получает готовых правил, а выводит их сам, анализируя множество примеров.

От распознавания картинок до разговора

Долгое время нейросети (так называют математические модели, вдохновлённые устройством человеческого мозга) умели только узкие вещи: отличать кошку от собаки, распознавать рукописные цифры на конверте, предлагать следующее слово при наборе SMS.

Прорыв случился, когда исследователи научились обучать модели на огромных объёмах текста — по сути, на большей части текстов, которые вообще существуют в открытом доступе в интернете: книгах, статьях, форумах, документации к программам. Модель училась не «понимать» текст в человеческом смысле, а предсказывать, какое слово должно идти следующим, с учётом всего, что было написано раньше. Оказалось, что если делать это в невероятно больших масштабах — миллиарды и триллионы примеров, — модель начинает демонстрировать что-то похожее на рассуждение, способность отвечать на вопросы, писать код, переводить языки и поддерживать связный диалог.

Такие модели называют **большими языковыми моделями**, или **LLM** (Large Language Model — большая языковая модель). Claude, о котором пойдёт речь в этой книге, — это именно такая модель, созданная компанией Anthropic.

ТЕРМИН: LLM (большая языковая модель) — нейросеть, обученная на огромном количестве текста, которая умеет генерировать связный текст, отвечать на вопросы и выполнять множество текстовых задач: от перевода до написания программного кода.

Значит ли это, что ИИ «думает»?

Здесь стоит остановиться и внести ясность, потому что вокруг этой темы много путаницы и маркетинговых преувеличений.

Языковая модель не обладает сознанием, не «понимает» текст так, как понимаете его вы, и не имеет собственных желаний или целей в человеческом смысле. По сути, это чрезы-

чайно сложная система предсказания текста, которая, тем не менее, обучена таким образом, что результат её работы очень похож на осмысленное рассуждение, а во многих практических задачах — не хуже, а порой и лучше, чем у человека.

Для наших с вами практических целей — использования Claude Code для написания программ — совершенно не важно, философский это вопрос «мыслит ли машина». Важно другое: такая модель умеет читать код, писать код, находить в нём ошибки, объяснять, что делает та или иная строчка, и делать всё это в диалоге с вами, реагируя на ваши уточнения и пожелания. Этого более чем достаточно, чтобы такой инструмент стал по-настоящему полезным помощником.

ВНИМАНИЕ: Языковая модель может ошибаться и иногда «придумывает» факты, которых не существует (это явление называют галлюцинацией модели). При работе с кодом это тоже случается: модель может предложить несуществующую функцию или библиотеку. Об этом мы подробно поговорим в главе про безопасность и хорошие привычки.

Зачем ИИ вообще учат писать код

Написание программного кода — на первый взгляд, очень специфичная и «человеческая» задача, требующая логики, точности и творчества одновременно. Но на самом деле код — это тоже текст, написанный по строгим правилам (правила языка программирования), и в интернете накоплены гигантские объёмы кода: миллионы открытых проектов, документации, вопросов и ответов на форумах программистов.

Обучив модель на этом массиве, исследователи получили ИИ, который способен не просто дописывать код по шаблону, а понимать задачу, сформулированную на обычном человеческом языке, и превращать её в работающую программу. Например, вы можете написать: «Сделай страницу, где будет форма с именем и почтой, и при отправке форма показывает надпись “Спасибо!”» — и получить готовый работающий код.

Это открывает удивительную возможность: программировать теперь может человек, который никогда не изучал языки программирования формально. Не потому что программирование стало не нужно — а потому что появился переводчик между человеческим языком, на котором мы формулируем задачи, и языком кода, на котором эти задачи выполняет компьютер.

Именно этот переводчик — с некоторыми важными оговорками, о которых мы поговорим ниже, — и есть Claude Code.

Зачем тогда вообще эта книга, если ИИ и так всё сделает?

Хороший вопрос, и на него стоит ответить честно.

Во-первых, чтобы эффективно объяснить компьютеру задачу, полезно понимать хотя бы в общих чертах, как устроены программы, файлы и папки, что такое сайт и чем он отличается от приложения. Это как с автомобилем: вам не обязательно быть автомехаником, чтобы водить машину, но полезно знать, где находится руль, педаль тормоза и что означает загоревшаяся лампочка на панели.

Во-вторых, ИИ не работает как волшебная кнопка «сделай хорошо». Он гораздо эффективнее, если вы умеете правильно формулировать задачи, разбивать большую цель на маленькие шаги и проверять результат. Этому тоже нужно учиться, и как раз этому посвящена значительная часть книги.

В-третьих, важно понимать границы возможностей ИИ: что он делает отлично, где регулярно ошибается, и какие решения всё равно должен принимать человек. Слепое доверие к любым предложениям ИИ — плохая идея, и мы подробно разберём, как этого избежать.

Небольшое резюме главы

- Искусственный интеллект в широком смысле — это программы, которые учатся на примерах, а не работают по жёстко прописанным правилам.
- Большие языковые модели (LLM), такие как Claude, обучены на огромных объёмах текста и умеют вести диалог, отвечать на вопросы и писать код.
- Языковая модель не обладает сознанием, но её практическая польза от этого не становится меньше.
- Понимание базовых принципов работы ИИ и программирования делает работу с такими инструментами, как Claude Code, значительно эффективнее.

В следующей главе мы познакомимся конкретно с Claude — моделью, которая стоит за Claude Code, — и разберёмся, чем она отличается от других известных вам чат-ботов.

Глава 2. Что такое Claude и чем он отличается от других ИИ

Кто такая компания Anthropic

Claude — это семейство больших языковых моделей, которое разрабатывает компания Anthropic, основанная в 2021 году бывшими сотрудниками OpenAI (компания, создавшей ChatGPT). Основатели Anthropic ушли, чтобы заниматься разработкой ИИ с особым упором на безопасность и предсказуемость поведения модели — отсюда, кстати, и название: Anthropic происходит от слова anthropic — «имеющий отношение к человеку».

Название «Claude» модель получила в честь Клода Шеннона — американского математика и инженера, которого называют отцом теории информации: именно он в середине XX века заложил математические основы того, как информация вообще может передаваться, храниться и обрабатываться. В каком-то смысле это символическое имя для программы, которая работает с языком и информацией на невероятном для человека масштабе.

Чем Claude отличается от других языковых моделей

Если вы уже пользовались похожими продуктами — например, ChatGPT от OpenAI или Gemini от Google, — у вас, возможно, есть вопрос: а зачем нужен ещё один чат-бот, если уже есть привычный?

На практике все крупные современные языковые модели умеют похожие вещи: отвечать на вопросы, писать тексты, помогать с кодом. Различия между ними — это как различия между хорошими автомобилями разных марок: у каждой свой характер, сильные стороны и нюансы.

Несколько особенностей, которые в сообществе разработчиков принято выделять у Claude:

Работа с кодом. Модели семейства Claude на протяжении последних версий стабильно показывают одни из лучших результатов именно в задачах программирования — согласно независимым тестам, которые измеряют способность модели решать реальные инженерные задачи в открытых проектах. Именно поэтому вокруг Claude выросла целая экосистема инструментов для разработчиков, и Claude Code — один из них.

Способность работать с длинным контекстом. Claude умеет удерживать в «памяти» одновременно очень большой объём текста — это может быть, например, целая книга или крупный программный проект из десятков файлов — и рассуждать, учитывая всю эту информацию целиком, а не только последние несколько сообщений диалога.

Подход к безопасности. Anthropic изначально позиционирует себя как компания, для которой безопасность и предсказуемость ИИ — не побочная, а центральная задача. На практике это выражается, например, в том, что Claude обучен избегать помощи в создании по-настоящему опасных вещей (вредоносных программ, оружия и так далее) и стараться быть честным, в том числе признавать, когда он чего-то не знает или может ошибаться.

СОВЕТ: Не относитесь к любой модели ИИ как к абсолютному авторитету. Даже самая совершенная система может ошибаться. Хорошая привычка — воспринимать ответы ИИ как мнение очень начитанного, но иногда невнимательного коллеги, которого стоит перепроверять в важных вопросах.

Claude как чат и Claude как инструмент внутри программ

Важно понимать, что «Claude» — это не одна конкретная кнопка или программа, а технология, которую можно использовать по-разному:

-

Claude.ai

— обычный сайт и приложение для переписки с Claude, похожее на привычный чат-бот: вы печатаете вопрос, получаете ответ.

-

API Claude

— способ, которым разработчики встраивают Claude в свои собственные программы и сервисы. Например, когда какое-нибудь приложение для заметок предлагает вам «улучшить текст с помощью ИИ», очень вероятно, что за этой кнопкой стоит именно обращение к модели вроде Claude.

-

Claude Code

— инструмент, о котором эта книга, специально созданный для того, чтобы Claude мог не просто разговаривать о коде, а напрямую работать с файлами на вашем компьютере: создавать их, редактировать, запускать программы и проверять результат.

Иными словами, Claude Code — это Claude, которому дали «руки»: возможность не просто советовать, что нужно написать в файле, а самому открывать этот файл, вносить в него изменения и смотреть, что получилось.

Что значит «Claude Code» дословно

Название говорящее: **Code** по-английски означает «код», то есть программный код — набор инструкций, написанных на специальном языке, который понимает компьютер. Claude Code — это программа, которая устанавливается на ваш компьютер и запускается через терминал (о том, что это такое, мы подробно поговорим в четвёртой главе), и внутри которой вы общаетесь с Claude на обычном языке, а Claude в ответ создаёт и изменяет настоящие файлы в вашем проекте.

Представьте помощника, который сидит рядом с вами за одним компьютером. Вы говорите: «Сделай так, чтобы кнопка стала зелёной», а он тут же открывает нужный файл, находит строчку с описанием цвета кнопки, меняет её и говорит: «Готово, я поменял цвет на зелёный, вот файл, в котором это сделано». В следующей главе мы разберём эту идею подробнее — и что конкретно можно поручать такому помощнику.

А это точно не опасно — доверять программе доступ к моим файлам?

Резонный вопрос, и мы вернёмся к нему подробно в главе о безопасности. Пока коротко: Claude Code по умолчанию работает только с той папкой проекта, в которой вы его запустили, и перед выполнением потенциально рискованных действий (например, удаление файлов или выполнение неизвестной команды) обычно спрашивает у вас подтверждение. Это устроено похоже на то, как антивирус или операционная система спрашивает «разрешить этому приложению доступ к файлам» — финальное решение всегда остаётся за вами.

Небольшое резюме главы

- Claude — семейство языковых моделей от компании Anthropic, известное, среди прочего, высокими результатами в задачах программирования.
- Claude доступен в разных формах: как обычный чат (Claude.ai), как технология внутри сторонних приложений (через API) и как специализированный инструмент для программирования — Claude Code.
- Claude Code даёт модели возможность не просто советовать, а напрямую работать с файлами вашего проекта — создавать, изменять и проверять код.
- Ни одна модель ИИ не застрахована от ошибок, и здоровая привычка — перепроверять важные решения, а не слепо доверять любому ответу.

В следующей главе мы разберём, что конкретно умеет Claude Code, зачем вообще нужен ИИ-помощник, который пишет код, и какие задачи ему стоит, а какие не стоит поручать.

Глава 3. Знакомство с Claude Code: зачем нужен ИИ-помощник, который пишет код

Что вообще значит «программировать»

Прежде чем разбирать, что умеет Claude Code, полезно на пальцах понять, чем вообще занимается программист.

Компьютер — это, по сути, очень быстрый и очень буквальный исполнитель инструкций. Он не умеет догадываться и не понимает «примерно то, что вы имели в виду». Программирование — это процесс записи задачи в виде настолько точных и однозначных инструкций, что компьютер способен их выполнить, ни разу не «задумавшись» и не переспросив.

Эти инструкции записываются на **языках программирования** — специальных искусственных языках вроде Python, JavaScript, HTML — у каждого свои правила и своя область применения. Совокупность таких инструкций, записанных в файлы, и называется программным кодом, или просто кодом.

Традиционно, чтобы создать сайт, приложение или программу, человек должен был выучить синтаксис такого языка (то есть его грамматику), запомнить сотни правил и команд, и затем вручную писать код строчка за строчкой, попутно разбираясь с ошибками — а они возникают постоянно, даже у опытных программистов.

Что меняет Claude Code

Claude Code меняет саму механику этого процесса. Вместо того чтобы самому вспоминать точный синтаксис нужной команды, вы объясняете задачу простыми словами, а Claude Code превращает это объяснение в реальный, работающий код.

Приведу конкретный пример диалога, каким он мог бы выглядеть на практике:

Вы: Хочу сделать простую страницу-визитку с моим именем, короткой фразой о себе и кнопкой “Написать мне”, которая открывает почту.

Claude Code: Хорошо, я создам файл index.html с базовой структурой страницы: заголовков с именем, абзац с описанием и кнопку-ссылку mailto. Также добавлю немного стилей, чтобы страница выглядела аккуратно. Создаю файл...

После этого в вашей папке появится настоящий работающий файл, который можно открыть в браузере и увидеть результат. Если что-то не понравилось — например, хочется другой цвет фона, — вы просто пишете об этом следующим сообщением, и Claude Code вносит изменения в уже существующий файл.

Это принципиально другой способ работы по сравнению с обычным чатом вроде ChatGPT или Claude.ai, где вам бы просто показали текст кода, а копировать его в файлы, сохранять и запускать пришлось бы самостоятельно. Claude Code делает всё это за вас — но объясняет, что и почему делает, на каждом шаге.

Из чего складывается «программа» на практике

Чтобы понимать, что вообще происходит, когда Claude Code говорит «я создал файл», полезно уже сейчас (мы вернёмся к этому подробнее в седьмой главе) представлять общую картину.

Любой сайт, приложение или программа — это набор текстовых файлов, лежащих в папке на компьютере, каждый из которых отвечает за свою часть работы. Например, простой

сайт может состоять из файла со структурой страницы (какой текст где расположен), файла с оформлением (какие цвета, шрифты, отступы) и файла с логикой поведения (что происходит при нажатии на кнопку). Claude Code умеет создавать все эти файлы, редактировать их по вашей просьбе и следить за тем, чтобы они правильно работали вместе.

ТЕРМИН: Проект — общее название для папки со всеми файлами, из которых состоит ваша программа или сайт.

Что Claude Code умеет делать конкретно

Перечислим основные возможности, чтобы у вас сложилась целостная картина. Каждую из них мы подробно разберём на практике в четвёртой части книги.

Создание кода с нуля. Вы описываете, что хотите получить, — от простой страницы до небольшого приложения, — и Claude Code создаёт нужные файлы и пишет в них код.

Редактирование существующего кода. Если у вас уже есть проект (например, вы скачали шаблон сайта или продолжаете старую работу), Claude Code может его открыть, разобраться, как он устроен, и вносить в него изменения по вашему запросу.

Поиск и исправление ошибок. Программы часто не работают с первого раза — это нормально даже для профессионалов. Claude Code умеет читать сообщения об ошибках (которые для новичка обычно выглядят как нечитаемая абракадабра), находить их причину в коде и предлагать или сразу вносить исправление.

Объяснение кода. Если вы наткнулись на чужой код или на код, написанный самим Claude Code некоторое время назад, и не помните, как он работает, можно попросить объяснить его простыми словами, строчка за строчкой.

Выполнение команд. Помимо редактирования файлов, Claude Code может выполнять и другие технические действия: устанавливать нужные для проекта вспомогательные библиотеки, запускать программу для проверки, работать с системой контроля версий git (о ней поговорим в девятой главе).

Работа с несколькими файлами одновременно. Реальные проекты часто состоят из десятков и сотен файлов. Claude Code умеет ориентироваться в такой структуре, находить нужное место для изменений и следить, чтобы правки в одном файле не сломали работу другого.

Чего Claude Code не умеет и не должен делать

Важно сразу расставить честные ожидания, чтобы не разочароваться и не совершить ошибок.

Claude Code — не телепат. Чем более расплывчато сформулирована задача, тем более непредсказуемым будет результат. «Сделай мне красивый сайт» — это плохая постановка задачи. «Сделай сайт-визитку с фотографией, тремя абзацами текста о моём опыте работы и формой обратной связи внизу страницы» — намного лучше. Мы посвятим отдельную главу тому, как правильно формулировать задачи (это называется промптингом).

Claude Code не заменяет полностью понимание того, что вы делаете. Даже с самым лучшим ИИ-помощником вам стоит хотя бы в общих чертах понимать получившийся результат, иначе вы рискуете оказаться в ситуации, когда программа что-то делает не так, а вы даже не знаете, где искать причину. Именно поэтому в этой книге так много внимания уделяется базовым понятиям программирования.

Claude Code может ошибаться, особенно в сложных, объёмных или нестандартных задачах. Как и любая модель ИИ, он может предложить код с ошибкой, использовать несуществующую функцию или неправильно понять задачу. Хорошая практика — всегда проверять резуль-

тат, особенно перед тем как публиковать сайт в открытом доступе или отправлять программу другим людям.

Claude Code — не замена специалисту в сложных, ответственных или требующих особой квалификации задачах: медицинское ПО, финансовые системы, системы безопасности. Для учебных задач, личных проектов, автоматизации своей рутины — прекрасный инструмент. Для критически важных систем всегда нужна проверка профессионалом.

Кому и зачем это может пригодиться

Стоит развеять миф, что такие инструменты нужны только тем, кто «хочет стать программистом». На практике Claude Code полезен гораздо более широкому кругу людей:

- Человеку, который хочет сделать личный сайт-портфолио или визитку, не тратя деньги на веб-студию.
- Небольшому предпринимателю, которому нужен простой скрипт для автоматической обработки таблицы с заказами.
- Студенту, который изучает программирование и хочет иметь под рукой терпеливого репетитора, готового объяснить одно и то же понятие десятью разными способами.
- Человеку творческой профессии, который хочет собрать небольшой интерактивный проект — например, генератор случайных идей или личный дневник с красивым оформлением.
- Просто любопытному человеку, который хочет понимать, как устроен цифровой мир вокруг, и однажды сказать: «Да, я сам написал программу, которая реально работает».

Небольшое резюме главы

- Программирование — это запись задачи в виде точных инструкций для компьютера на специальном языке.
- Claude Code превращает описание задачи на обычном языке в реальный работающий код, самостоятельно создавая и редактируя файлы.
- Инструмент умеет писать код с нуля, редактировать существующий, находить и чинить ошибки, объяснять код и выполнять технические команды.
- Claude Code не заменяет понимание сути происходящего и может ошибаться — результат всегда стоит проверять.
- Инструмент полезен не только будущим программистам, а практически любому человеку с идеей небольшого цифрового проекта.

Первая часть книги дала нам общую картину. Теперь пора переходить от теории к практике — и первый практический шаг будет заключаться в том, чтобы установить Claude Code на ваш компьютер. Но перед этим нужно разобраться с инструментом, без которого не обойтись, — терминалом. Этому посвящена следующая, четвёртая глава.

Часть 2. Установка и настройка

Глава 4. Что такое терминал и командная строка

Почему нельзя просто скачать программу с сайта и открыть двойным кликом

Большинство привычных нам программ — браузер, текстовый редактор, плеер — устанавливаются одинаково: скачиваем файл-установщик, дважды кликаем, следуем подсказкам мастера установки, готово. У Claude Code (и у подавляющего большинства инструментов для разработчиков) основной способ установки и запуска другой — через терминал.

Дело в том, что инструменты для разработки создаются в первую очередь для людей, которые и так весь день работают в терминале, поэтому для них он гораздо удобнее и быстрее графических окошек с кнопками. Вам же, чтобы воспользоваться Claude Code, придётся освоить самый необходимый минимум работы с этим инструментом — и это, обещаю, гораздо проще, чем кажется на первый взгляд.

Что такое терминал

Терминал (его также называют консолью или командной строкой) — это программа, которая позволяет управлять компьютером не с помощью мышки и кнопок, а с помощью текстовых команд, которые вы печатаете и подтверждаете нажатием клавиши Enter.

ТЕРМИН: Терминал — программа для ввода текстовых команд, с помощью которых можно управлять компьютером: открывать папки, запускать программы, копировать и удалять файлы и многое другое.

Представьте, что вместо того, чтобы искать нужную папку через проводник, кликая по иконкам, вы просто печатаете: «покажи мне, что лежит в папке Документы» — только на специальном языке команд, который понимает компьютер. Именно так и работает терминал.

На первый взгляд это может показаться шагом назад — зачем печатать команды, если можно кликать мышкой? Но на практике у терминала есть важные преимущества: команды можно быстро повторять, комбинировать друг с другом, автоматизировать, а главное — почти все инструменты разработки (включая Claude Code) устроены именно как программы, управляемые через терминал.

Где найти терминал на вашем компьютере

Терминал уже установлен на любом компьютере — Windows, macOS или Linux, — искать и скачивать отдельно ничего не нужно.

На Windows нужная программа называется «Терминал» (в новых версиях Windows 10 и 11) или PowerShell. Найти её проще всего так: нажмите на клавиатуре клавишу Windows (с логотипом Windows), начните печатать слово «терминал» и нажмите Enter, когда увидите нужный результат в списке.

На macOS программа называется Terminal (Терминал). Найти её можно через Spotlight — нажмите одновременно клавиши Cmd и Пробел, напечатайте «Terminal» и нажмите Enter. Также её можно найти в папке Программы → Утилиты.

На Linux способ открытия терминала зависит от конкретной версии системы, но почти всегда работает сочетание клавиш `Ctrl + Alt + T`, либо нужную программу можно найти в списке приложений по слову «Terminal» или «Консоль».

СОВЕТ: Закрепите иконку терминала на панели задач или в доке — в ближайшие недели вы будете открывать его очень часто.

Первое знакомство: не бойтесь чёрного окна

Когда вы откроете терминал, вы увидите окно (часто чёрное или тёмное, хотя цвет можно менять) с мигающим курсором и небольшой надписью перед ним — это называется **приглашением командной строки**, оно показывает, что программа готова принять от вас команду.

Не переживайте — сломать что-то, просто печатая команды и с любопытством нажимая `Enter`, крайне сложно. Терминал не выполняет ничего разрушительного без явного запроса, а большинство команд просто показывают информацию, ничего не удаляя и не меняя.

Давайте попробуем несколько самых безобидных и полезных команд, чтобы почувствовать, как это работает. Впечатайте команду и нажмите `Enter` — программа выполнит её и покажет результат под ней.

Узнать, в какой папке вы сейчас находитесь.

На Windows (PowerShell):

```
pwd
```

На macOS и Linux:

```
pwd
```

Эта команда (`pwd` расшифровывается как `print working directory` — «показать текущую рабочую папку») покажет путь к папке, в которой сейчас «находится» терминал.

ТЕРМИН: Рабочая папка (или текущая директория) — папка, с содержимым которой в данный момент работает терминал. Все команды, которые вы вводите, по умолчанию применяются именно к ней.

Посмотреть, что лежит в этой папке.

На Windows:

```
dir
```

На macOS и Linux:

```
ls
```

Вы увидите список файлов и папок — то же самое, что вы обычно видите в проводнике или Finder, только в виде текста.

Перейти в другую папку.

Команда одинаковая на всех системах:

```
cd Название_папки
```

`cd` расшифровывается как `change directory` — «сменить папку». Например, чтобы перейти в папку Документы (Documents), нужно написать:

```
cd Documents
```

Чтобы вернуться на уровень выше (в родительскую папку), используется специальное обозначение из двух точек:

```
cd ..
```

СОВЕТ: Если название папки состоит из нескольких слов с пробелами, заключите его в кавычки: `cd "Мои документы"`.

Вернуться в домашнюю папку пользователя.

На macOS и Linux:

```
cd ~
```

На Windows (PowerShell):

`cd ~`

Символ `~` (тильда) — это сокращение для домашней папки текущего пользователя, где обычно находятся ваши личные файлы, документы и рабочий стол.

Небольшой словарь для начала работы

Прежде чем двигаться дальше, зафиксируем несколько понятий, которые будут постоянно встречаться в книге.

Путь (path) — адрес файла или папки на компьютере, записанный в виде текста. Например, `C:\Users\Анна\Documents\Проект` на Windows или `/Users/anna/Documents/Проект` на macOS — это путь к папке с проектом.

Команда — текстовая инструкция, которую вы вводите в терминал, чтобы он что-то сделал.

Аргумент — дополнительное слово или значение после самой команды, уточняющее, что именно нужно сделать. Например, в команде `cd Documents` слово `Documents` — это аргумент, указывающий, в какую именно папку нужно перейти.

Флаг (или опция) — специальный аргумент, обычно начинающийся с одного или двух дефисов (например, `-l` или `--help`), который меняет поведение команды. Мы будем встречать флаги при установке и настройке Claude Code.

Опечатки — это нормально

Если вы неправильно напечатали команду, терминал обычно вежливо сообщит об ошибке текстом вроде `command not found` («команда не найдена») — и ничего страшного не произойдёт. Просто исправьте опечатку и попробуйте снова. Работа с терминалом на первых порах — это в том числе привыкание к тому, что ошибки — это нормальная, рутинная часть процесса, а не повод для паники.

ВНИМАНИЕ: Единственное, о чём действительно стоит быть аккуратным в терминале, — это команды удаления файлов (например, `rm` на macOS/Linux или `del` на Windows). В отличие от удаления через проводник, файлы, удалённые через терминал, обычно не попадают в корзину, а удаляются безвозвратно. В рамках этой книги мы почти не будем использовать такие команды напрямую — Claude Code сам будет запрашивать подтверждение перед удалением чего-либо.

Небольшое резюме главы

- Терминал — программа для управления компьютером с помощью текстовых команд, установлена по умолчанию на всех системах.

-

Основные команды для начала:

`pwd`

(где я нахожусь),

`ls`

`/`

`dir`

(что лежит в папке),

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.