

*Посвящается моей маме, Сильвии,  
которая радовалась каждой моей победе  
и помогала мне переживать каждое поражение*

# СОДЕРЖАНИЕ

<i>От издательства</i> .....	10
<i>Благодарности</i> .....	12
<i>Введение</i> .....	14

## **Часть I ИНТЕРПРЕТАТОРЫ**..... 19

### **Глава 1. Минимально возможный язык программирования**..... 20

Что такое Brainfuck?.....	20
Что определяет полноту языка по Тьюрингу?.....	21
Как работает Brainfuck.....	22
Структура интерпретатора.....	27
Реализация Brainfuck на Python.....	28
Получение исходного файла.....	29
Создание интерпретатора.....	29
Запуск интерпретатора.....	33
Тестирование интерпретатора.....	34
Практические приложения.....	37
Упражнения.....	38

### **Глава 2. Создание интерпретатора языка BASIC**..... 39

Основы NanoBASIC.....	40
История BASIC.....	40
Парадигма, синтаксис и семантика NanoBASIC.....	41
Операторы GOTO, GOSUB и RETURN.....	45
Стиль и особенности NanoBASIC.....	46
Пример программы NanoBASIC.....	47
Формализация синтаксиса NanoBASIC.....	48
Реализация NanoBASIC.....	53
Токенизатор.....	54

Узлы.....	58
Ошибки .....	61
Парсер.....	63
Среда выполнения.....	72
Запуск программы .....	77
Тестирование NanoBASIC.....	77
Практические приложения .....	82
Упражнения .....	83
<b>Часть II ИСКУССТВО И ВЫЧИСЛЕНИЯ.....</b>	<b>84</b>
<b>Глава 3. Ретрообработка изображений .....</b>	<b>85</b>
Что такое дизеринг? .....	85
Начало работы.....	88
Алгоритм дизеринга.....	90
Файловый формат MacPaint .....	95
Преобразование байтов в биты.....	96
Реализация кодирования по длине последовательности.....	98
Тестирование кодирования по длине последовательности.....	103
Преобразование в MacBinary .....	104
Сведение всего воедино.....	108
Результаты .....	109
Практические приложения.....	112
Упражнения .....	113
<b>Глава 4. Стохастический алгоритм живописи .....</b>	<b>114</b>
Как это работает .....	114
Опции командной строки .....	120
Формат SVG.....	121
Алгоритм .....	124
Основная реализация.....	125
Настройка .....	125
Вспомогательные методы .....	128
Пробы .....	129
Вывод изображения .....	132
Результаты .....	134
Практические приложения.....	138
Упражнения .....	139
<b>Часть III ЭМУЛЯТОРЫ .....</b>	<b>141</b>
<b>Глава 5. Создание виртуальной машины CHIP-8 .....</b>	<b>142</b>
Виртуальные машины.....	142
Виртуальная машина CHIP-8.....	144

Регистры и память.....	145
Инструкции .....	146
Реализация.....	150
Цикл выполнения .....	151
Аргументы командной строки .....	154
Настройка VM и вспомогательные функции .....	154
Графика .....	157
Выполнение инструкций .....	160
Тестирование VM.....	166
Запуск игр.....	167
Практические приложения .....	169
Упражнения .....	170
<b>Глава 6. Эмуляция игровой консоли NES.....</b>	<b>172</b>
О платформе NES .....	173
Аппаратная платформа.....	174
Программное обеспечение .....	176
Создание эмулятора .....	177
Планирование структуры .....	177
Создание главного цикла.....	178
Эмуляция картриджа .....	181
Эмуляция центрального процессора.....	185
Принципы работы PPU.....	212
Реализация PPU.....	222
Тестирование эмулятора .....	233
Игровой процесс.....	236
Практические приложения .....	240
Упражнения .....	241
<b>Часть IV  ОЧЕНЬ ПРОСТОЕ МАШИННОЕ ОБУЧЕНИЕ.....</b>	<b>242</b>
<b>Глава 7. Классификация с помощью k-ближайших соседей.....</b>	<b>243</b>
Становление машинного обучения .....	243
Как работает KNN .....	245
Реализация классификации с помощью KNN .....	247
Классификация рыб .....	250
Классификация рукописных цифр .....	254
Практические приложения .....	258
Упражнения .....	259
<b>Глава 8. Регрессия с помощью k ближайших соседей .....</b>	<b>260</b>
Как работает регрессия KNN.....	260
Реализация регрессии с помощью KNN.....	262
Прогнозирование веса рыб .....	263

Прогнозирование недостающей части рукописной цифры .....	264
Практические приложения .....	268
Упражнения .....	269
<b>Послесловие</b> .....	270
Что мы сделали и что будет дальше .....	270
Об изучении компьютерных технологий .....	271
Интерпретаторы .....	273
Компьютерное изобразительное искусство .....	273
Эмуляторы .....	274
Машинное обучение .....	274
<b>Приложение. Побитовые операции</b> .....	276
Обзор двоичной системы счисления .....	276
Распространенные побитовые операции .....	278
Сдвиг влево (<<) .....	278
Сдвиг вправо (>>) .....	279
ИЛИ ( ) .....	280
И (&) .....	281
XOR (^) .....	282
Дополнение (~) .....	283
<b>Предметный указатель</b> .....	284

# ОТ ИЗДАТЕЛЬСТВА

## Отзывы и пожелания

Мы всегда рады отзывам наших читателей. Расскажите нам, что вы думаете об этой книге – что понравилось или, может быть, не понравилось. Отзывы важны для нас, чтобы выпускать книги, которые будут для вас максимально полезны.

Вы можете написать отзыв на нашем сайте [www.dmkpress.com](http://www.dmkpress.com), зайдя на страницу книги и оставив комментарий в разделе «Отзывы и рецензии». Также можно послать письмо главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com); при этом укажите название книги в теме письма.

Если вы являетесь экспертом в какой-либо области и заинтересованы в написании новой книги, заполните форму на нашем сайте по адресу [http://dmkpress.com/authors/publish\\_book/](http://dmkpress.com/authors/publish_book/) или напишите в издательство по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

## Список опечаток

Хотя мы приняли все возможные меры для того, чтобы обеспечить высокое качество наших текстов, ошибки все равно случаются. Если вы найдете ошибку в одной из наших книг, мы будем очень благодарны, если вы сообщите о ней главному редактору по адресу [dmkpress@gmail.com](mailto:dmkpress@gmail.com). Сделав это, вы избавите других читателей от недопонимания и поможете нам улучшить последующие издания этой книги.

## Нарушение авторских прав

Пиратство в интернете по-прежнему остается насущной проблемой. Издательство «Books.kz» очень серьезно относится к вопросам защиты авторских прав и лицензирования. Если вы столкнетесь в интернете с незаконной публикацией какой-либо из наших книг, пожалуйста, пришлите нам ссылку на интернет-ресурс, чтобы мы могли применить санкции.

Ссылку на подозрительные материалы можно прислать по адресу электронной почты [dmkpress@gmail.com](mailto:dmkpress@gmail.com).

Мы высоко ценим любую помощь по защите наших авторов, благодаря которой мы можем предоставлять вам качественные материалы.

## Об авторе

**Дэвид Копек** (David Kores) – доцент кафедры информатики в колледже Олбрайт. Он присоединился к академическому сообществу в 2016 году после работы в качестве разработчика программного обеспечения со специализацией в области создания приложений для iOS. Является автором четырех технических книг, в том числе «*Классические проблемы информатики на Python*» (*Classic Computer Science Problems in Python*), которая была переведена на восемь языков и опубликована по всему миру. Копек – активный разработчик приложений и подкастер. Живет с женой и тремя детьми в городе Вайомиссинг, штат Пенсильвания, США.

## О техническом рецензенте

**Майкл Кеннеди** (Michael Kennedy) хорошо известен среди специалистов по Python благодаря своей работе над подкастами *Talk Python to Me* и *Python Bytes*, в которых на протяжении почти 10 лет освещаются важные темы и новости сообщества Python. Является основателем курсов Talk Python Training, где предлагается множество онлайн-курсов для разработчиков, а также членом Python Software Foundation. Кеннеди живет в Портленде, штат Орегон, США. Когда он не занимается программированием и не проводит время с семьей, его можно увидеть путешествующим на мотоцикле по местным горам.

# БЛАГОДАРНОСТИ

Прежде всего я хотел бы поблагодарить вас, читатель, за приобретение этой книги. Вероятно, вы могли бы найти в интернете учебные материалы по большинству тем, затронутых в этой книге, но вряд ли они будут настолько продуманы, согласованы и тщательно оформлены. По крайней мере на мой взгляд, и именно поэтому я написал эту книгу. Приобретая ее, вы поддержали не только ее создание, но и дальнейшее существование всей отрасли, выпускающей подобные книги.

Далее я хотел бы поблагодарить колледж Шамплейн, который предоставил мне творческий отпуск после семи лет работы. Осталось не так много отраслей, где сотрудникам дают почти год отпуска, чтобы они могли заниматься своими профессиональными делами, и при этом еще и платят им за это. Иногда из этого времени и пространства рождается что-то замечательное. Написание данной книги было моим проектом на период этого творческого отпуска.

Я хотел бы поблагодарить свою семью, которая поддерживала меня в работе над этим проектом, особенно мою маму Сильвию и мою жену Ребекку. Мои дети, Дэниел, Вера и Люсиль, еще слишком малы, чтобы по-настоящему осознавать, что значит писать техническую книгу, но я не смог бы ее написать, если бы Ребекка не заботилась о них, поэтому вместо того, чтобы благодарить их, я поблагодарю ее во второй раз.

Я хотел бы поблагодарить издательство No Starch Press за то, что они поверили в эту книгу. Я пришел к ним с уже полностью написанным черновым вариантом рукописи, что является немного необычным для технической книги, и они поверили в ее содержание настолько, что вложили ресурсы в ее доработку ради вашего удобства. В частности, я хотел бы поблагодарить Натана Хайдельбергера (Nathan Heidelberger), моего технического редактора, который нашел время, чтобы сопереживать читателям и найти как масштабные, так и небольшие возможности для улучшения книги. И конечно же, я хотел бы поблагодарить остальных членов команды No Starch Press, которые помогли

в подготовке и издании книги. Я также хотел бы поблагодарить моего технического рецензента Майкла Кеннеди (Michael Kennedy) за его ценные предложения.

И наконец, я хочу поблагодарить всех, кто оставил публичные отзывы о моих предыдущих книгах. Если бы вы не отреагировали положительно на мои предыдущие книги и не нашли время, чтобы это зафиксировать, эта книга никогда бы не получила такого импульса для старта. Если вы читаете это, пожалуйста, найдите время, чтобы оставить отзыв и об этой книге!

# ВВЕДЕНИЕ



Как устроен язык программирования? Как устроен простой компьютер? Я отношусь к тому типу людей, которые любят изучать новые предметы, начиная с самых основ, на практике. Мне нужно больше, чем просто общий обзор. Если и вы относитесь к такому типу людей, то вы нашли нужный ресурс. Благодаря семи проектам на Python, представленным в этой книге, вы сможете получить представление о некоторых фундаментальных принципах в области компьютерных наук.

## Для кого эта книга

Эта книга предназначена для программистов на языке Python среднего и продвинутого уровней. Если вы начинающий программист, вам, вероятно, стоит вернуться к этой книге позже. На протяжении всего текста подразумевается, что читатель знаком с синтаксисом и семантикой Python, умеет писать программы средней сложности, знает, как устанавливать библиотеки Python, и разбирается в базовых структурах данных, таких как списки, наборы и словари.

Несмотря на предположение о наличии у читателей некоторого опыта программирования, я не рассчитываю на их знания в области компьютерных технологий или высшей математики. Эта книга предназначена для тех, кто не имеет профильного образования в области вычислительной техники или хочет восполнить некоторые пробелы в своих знаниях. Например, если вы заинтересованы в написании собственного языка программирования, но никогда не посещали курсы по компиляторам, эта книга будет отличной отправной точкой. Если вы хотите написать эмулятор игровой консоли, эта книга подскажет вам, как это сделать. В ней также есть вполне понятное введение в тему машинного обучения.

Конкретные проекты, описанные в данной книге, могут не стать вашей конечной целью, но это и не требуется. Рассматривайте их как

инструмент для получения более глубоких знаний об алгоритмическом мышлении и принципах работы программного обеспечения, а также в качестве стартового этапа для ваших собственных исследований.

## О чем эта книга

Каждая глава представляет собой полноценный законченный проект, за исключением глав 7 и 8, которые вместе составляют один проект. Сложность представленных в книге семи проектов варьируется от легкой (интерпретатор Brainfuck в главе 1) до сложной (эмулятор NES в главе 6), но благодаря наличию исходного кода вы никогда не окажетесь в тупике и всегда сможете продолжить работу.

Каждый проект начинается с изложения определенной теоретической базы – достаточной для понимания того, что именно будет реализовано, без углубления в детали, а затем следует подробное описание кода. Главы также включают описание моих личных причин интереса к данной теме, обсуждение того, как реализуемые алгоритмы или вычислительные методы используются в реальной жизни, а также задачи для читателя с целью дальнейшего использования предоставленного кода.

Книга разделена на четыре части. Часть I посвящена изучению мира интерпретаторов путем написания реализаций двух простых языков программирования.

**Глава 1 «Минимально возможный язык программирования».** Brainfuck – это минималистичный язык программирования, который часто используется в образовательных целях из-за его простоты: весь язык состоит всего из восьми символов. Мы познакомимся с принципом работы очень простого интерпретатора, создадим свой собственный, который сможет запускать любую программу на Brainfuck. Мы также разберемся со значением термина «язык, полный по Тьюрингу».

**Глава 2 «Создание интерпретатора языка BASIC».** Язык программирования BASIC и его упрощенный вариант Tiny BASIC были популярны во времена компьютерной революции конца 1970-х годов. Мы разработаем интерпретатор для несколько упрощенного варианта Tiny BASIC под названием NanoBASIC. Это даст нам представление о компонентах более сложных интерпретаторов, в том числе о токенизаторе, парсере и среде выполнения.

В части II нам предстоит погрузиться в яркий мир компьютерного изобразительного искусства.

**Глава 3 «Ретрообработка изображений».** Когда технологии отображения были более простыми, для адаптации изображений к устройствам с ограниченной цветовой палитрой требовались алгоритмы дизеринга, то есть размытия (сглаживания). Мы разработаем

алгоритм дизеринга, способный отображать современные цветные фотографии на черно-белом экране оригинального Macintosh. Затем конвертируем полученные изображения в формат, совместимый с классическим приложением MacPaint, с использованием алгоритма сжатия с кодированием длины последовательности. Полученные изображения можно выводить на экран реальной системы Macintosh 1980-х годов.

**Глава 4 «Алгоритм стохастического раскрашивания».** Может ли сравнительно простой алгоритм помочь в создании сложного абстрактного искусства? Мы будем использовать стохастическую технику для генерации «впечатлений» из уже существующих изображений путем сопоставления случайных форм с исходным изображением. Затем разберемся с оптимизацией результатов с помощью алгоритма поиска максимума.

Часть III посвящена эмуляторам – программам, которые позволяют одним компьютерным системам имитировать работу других.

**Глава 5 «Создание виртуальной машины CHIP-8».** CHIP-8 – это спецификация виртуальной машины (Virtual Machine, VM), которая первоначально использовалась для разработки видеоигр в 1970-х годах. Создание виртуальной машины CHIP-8 обычно считается наиболее подходящим вариантом для первого знакомства с миром эмуляции: это относительно просто, но при этом включает в себя все этапы, необходимые для создания эмулятора. Наша виртуальная машина CHIP-8 будет способна запускать все игры CHIP-8, которые работали на системах в 1970-х годах.

**Глава 6 «Эмуляция игровой консоли NES».** NES была одной из самых продаваемых игровых консолей всех времен. Мы создадим эмулятор, который сможет запускать настоящие игры для NES. Он будет без звука, будет довольно медленным и не будет полностью точным или универсально совместимым, но все же станет отличным способом изучить не только эмуляторы, но и то, как работают компьютеры на низком уровне.

Наконец, часть IV представляет собой очень доступное введение в мир машинного обучения с использованием алгоритма  $k$ -ближайших соседей ( $k$ -nearest neighbors, KNN).

**Глава 7 «Классификация с помощью  $k$ -ближайших соседей».** Мы изучим KNN, который является, пожалуй, самым простым алгоритмом в машинном обучении (Machine Learning, ML), и задействуем его в качестве отправной точки для ознакомления с некоторыми вводными тезисами ML. Мы будем использовать KNN для классификации рыб, а также изображений рукописных цифр. Удивительно, но он справляется с последней задачей с точностью 98 %.

**Глава 8 «Регрессия с помощью  $k$  ближайших соседей».** Мы перейдем на следующий уровень KNN, используя его не только для классификации элементов по категориям, но и для прогнозирования неизвестных атрибутов точек данных. В конце главы с его помощью мы

будем предугадывать отсутствующие пиксели на изображении цифры, нарисованной пользователем.

В дополнение к основным главам в послесловии представлены некоторые рекомендуемые ресурсы для более глубокого изучения тем, затронутых в этой книге, а в приложении освещаются основы низкоуровневого оперирования битами в Python, что необходимо для реализации ряда проектов.

## Подход, принятый в этой книге

Я всегда стараюсь, чтобы мои книги были максимально лаконичными. Я ценю ваше время. Я использую формат, похожий на самоучитель, с упором на код, и по возможности даю коду возможность говорить самому за себя.

Это не учебник. Вы найдете в нем немного теории, особенно в начале каждой главы, но она никогда не затягивается, и вскоре мы переходим к коду. В книге содержится ровно столько информации, сколько нужно для понимания того, как работает каждый из проектов, и достаточно подсказок, чтобы вы были в курсе, где искать дополнительную информацию в случае возникновения желания углубиться в какую-либо из затронутых тем.

Я не утверждаю, что являюсь экспертом в области интерпретаторов, компьютерного изобразительного искусства, эмуляторов или машинного обучения. Это может показаться странным, когда это говорит автор книги по данным темам, но это действительно так. Я не эксперт, я учитель. Я работал разработчиком программного обеспечения и преподавателем по компьютерным технологиям в педагогическом колледже. Я утверждаю, что умею писать чистый код и объяснять его вам исключительно понятным образом. А поскольку я не эксперт, я не буду смотреть на вас свысока. Я буду относиться к вам как к равному, так как мы пройдем этот путь вместе. Это руководство, которое я хотел бы иметь под рукой в те времена, когда начинал самостоятельную работу над проектами в данных областях.

## О коде

Весь исходный код, используемый в этой книге, доступен в сопровождающем репозитории GitHub по адресу <https://github.com/davecom/ComputerScienceFromScratch>. Код был создан и протестирован с Python версий 3.12 и 3.13. Поскольку здесь используются некоторые функции Python 3.12, связанные с подсказками типов, часть кода не будет работать с более ранними версиями Python (но, вероятно, будет работать с любой новой версией Python для обозримого будущего). Тем не менее если удалить подсказки типов, большая часть кода будет работать с Python версии 3.10 и более поздними.

Я использовал подсказки типов Python (или «аннотации типов») во всем исходном коде, поскольку считаю, что они повышают удобочитаемость, указывая типы параметров и возвращаемых значений функции, без необходимости тщательного изучения кода или комментариев. Если они вам не нравятся, вы можете их игнорировать; они никак не влияют на работу кода. Я старался не злоупотреблять подсказками типов, так как некоторые считают их слишком многословными. Например, я редко использую их в телах функций, но использую в каждой сигнатуре функции. Я проверил весь исходный код на соответствие современной версии Pyright на момент написания книги.

В нескольких проектах этой книги используются внешние библиотеки. Вам необходимо установить Pygame, NumPy и Pillow в виртуальной среде, созданной для исходного кода книги, или в системном интерпретаторе Python. Для большинства читателей их установка не составит труда: достаточно выполнить команду `pip install pygame, numpy, pillow`. Файл *requirements.txt*, который может использовать менеджер пакетов `pip`, включен в репозиторий исходного кода книги.

## Исправления и комментарии

Репозиторий книги на GitHub – это отличное место, где можно оставить сообщение в случае обнаружения ошибки. Вы также можете связаться со мной по электронной почте [csfromscratch@oaksnow.com](mailto:csfromscratch@oaksnow.com) или через X [@davekopes](https://twitter.com/davekopes). Я буду рад вашим отзывам, как положительным, так и отрицательным. Если вам понравилась книга, пожалуйста, оставьте отзыв на Amazon или в том месте, где вы ее приобрели.

# **ЧАСТЬ I**

## **ИНТЕРПРЕТАТОРЫ**



Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

[e-Univers.ru](http://e-Univers.ru)