

Искусственный интеллект

ПРЕДЕЛЫ ВОЗМОЖНОГО

ARTIFICIAL UNINTELLIGENCE
HOW COMPUTERS MISUNDERSTAND THE WORLD

BROUSSARD MEREDITH

The MIT Press
Cambridge, Massachusetts
London, England

Мереди́т Бруссард

Искусственный интеллект

ПРЕДЕЛЫ ВОЗМОЖНОГО

Перевод с английского

АНО
АЛЬПИНА НОН-ФИКШН

Москва
2020

УДК 004.8
ББК 32.813
Б89

Переводчик Екатерина Арье
Научный редактор Сергей Марков
Редактор Антон Никольский

Бруссард М.

Б89 Искусственный интеллект: Пределы возможного / Мередит Бруссард ; Пер. с англ. — М. : Альпина нон-фикшн, 2020. — 362 с.

ISBN 978-5-00139-080-0

Книга авторитетного эксперта в области компьютерных технологий — призыв к здравомыслию. Всю свою сознательную жизнь Мередит Бруссард слышала, что технологии спасут мир, однако сегодня, продолжая восхищаться ими и участвовать в их создании, она относится к будущему не столь оптимистично. Всеобщий энтузиазм по поводу применения компьютерных технологий, по ее убеждению, уже привел к огромному количеству недоработанных решений в области проектирования цифровых систем. Выступая против техношовинизма и социальных иллюзий о спасительной роли технологий, Бруссард отправляется в путешествие по компьютерному миру: рискуя жизнью, садится за руль экспериментального автомобиля с автопилотом; задействует искусственный интеллект, чтобы выяснить, почему студенты не могут сдать стандартизованные тесты; использует машинное обучение, подсчитывая вероятность выживания пассажиров «Титаника»; как дата-журналист создает программу для поиска махинаций при финансировании кандидатов в президенты США. Только понимая пределы компьютерных технологий, утверждает Бруссард, мы сможем распорядиться ими так, чтобы сделать мир лучше.

УДК 82-84:004.8
ББК 84-49:32.813

Все права защищены. Никакая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, включая размещение в сети интернет и в корпоративных сетях, а также запись в память ЭВМ для частного или публичного использования, без письменного разрешения владельца авторских прав. По вопросу организации доступа к электронной библиотеке издательства обращайтесь по адресу mylib@alpina.ru

ISBN 978-5-00139-080-0 (рус.)
ISBN 978-0262038003 (англ.)

© Meredith Broussard, 2018
The rights to the Russian-language edition obtained through Alexander Korzhenevski Agency (Moscow)
© Издание на русском языке, перевод, оформление. ООО «Альпина нон-фикшн», 2020

Оглавление

I. Как работают компьютеры

1. Здравствуй, читатель.....	9
2. Hello, world!.....	25
3. Здравствуй, искусственный интеллект	53
4. Здравствуй, журналистика данных.....	69

II. Когда компьютеры не справляются

5. Почему ученики из малообеспеченных школ не справляются со стандартизированными тестами	83
6. Человеческий фактор.....	109
7. Машинное обучение: DL на ML	139
8. Эта машина сама не поедет	189
9. Популярное не значит хорошее.....	235

III. Сотрудничество

10. Автобус для стартаперов.....	257
11. Искусственный интеллект третьей волны	277
12. Стареющие компьютеры.....	305
Благодарности	317
Примечания.....	319
Библиография	331
Предметно-именной указатель	355

Как работают компьютеры

1

Здравствуй, читатель

Я люблю технологии. Любила их с тех самых пор, как родители купили мне, маленькой девочке, конструктор. Из маленьких металлических деталей мне предстояло собрать гигантского — как мне казалось — робота, работающего от мотора на батарейках. В детстве у меня было хорошее воображение, и я думала, что, как только соберу его, он сможет с легкостью передвигаться по дому, как и я, и станет мне новым другом. Тогда я бы научила его танцевать, а он приносил бы мне игрушки (не в пример моей собаке).

Сидя на красном шерстяном ковре на втором этаже, я проводила часы в мечтах за сборкой робота, затягивая множество винтов и болтов детским гаечным ключом. Самый волнительный момент настал, когда пришла пора устанавливать мотор. Накануне мы с мамой специально ездили в магазин, чтобы подобрать подходящие батарейки. Как только мы вернулись домой, я побежала наверх, чтобы подключить провода и наконец включить робота. Запуская механизм в надежде, что он изменит весь мир, я ощущала себя как братья Орвилл и Уилбур Райт.

Ничего не произошло.

Я проверила чертежи, несколько раз переключила рычаг вкл/выкл и перевернула батарейки. Ничего не происходило. Мой робот не работал. Я пошла за мамой.

— Поднимись, пожалуйста, наверх. Робот не работает, — грустно произнесла я.

— А ты пробовала выключить и включить его? — спросила мама.

— Да.

— А батарейки переворачивала? — продолжила она.

— Да! — раздражение нарастало. — Давай взглянем на него, — я взяла ее за руку и потащила наверх.

Она бегло изучила робота: проверила соединения проводов, несколько раз включала и выключала его.

— Он не работает, — заключила мама.

— Почему? — отозвалась я. Она могла просто сказать, что мотор сломан, но моя мама была сторонницей сложных объяснений. Она констатировала поломку мотора и затем рассказала про цепь оптовых поставок и конвейеры, а также напомнила мне, что я знаю, как работают фабрики, ведь мне нравилось смотреть, как огромные машины собирают коробки карандашей в телешоу «Улица Сезам».

Она объяснила: «Что-то может идти не по плану, когда создаешь вещи. И что-то пошло не так, когда они собирали этот мотор, в итоге он оказался в твоём наборе. А теперь нам нужно получить работающий мотор». Мы позвонили на горячую линию, по телефону, указанному в инструкции, и милые сотрудники компании выслали нам новый мотор по почте. Спустя примерно неделю мы его получили, и робот включился. Но я была разочарована. Он работал, но не так, как я ожидала, с трудом передвигался по нашему деревянному полу, застрял на ковре.

Кажется, он совсем не собирался становиться моим новым другом. Спустя несколько дней я разобрала его и приступила к сборке следующей модели из конструктора — колесу обозрения.

Я кое-чему научилась, собирая робота. Возясь с конструктором, я поняла, как с помощью инструментов можно собирать сложные механизмы и что это может быть увлекательно. Я узнала, что у меня действительно хорошее воображение, однако реалии технологий не всегда ему соответствуют. А еще я обнаружила, что детали могут быть неисправны.

Спустя несколько лет я стала учиться программированию и поняла, что опыт сборки робота из конструктора вполне соотносится с миром кодинга: я могла себе представить невероятно сложные программы, однако реальные возможности компьютеров чаще всего разочаровывали. Нередко программы просто не работали из-за неисправностей компьютера. Это меня не остановило, и я продолжаю создавать и восхищаться технологиями. У меня множество аккаунтов в социальных сетях. Однажды я хакнула мультиварку, чтобы подогреть 11 кг шоколада, — это было частью проекта. Я даже создала компьютеризированную систему автоматического полива в саду.

Однако с недавних пор я перестала верить в то, что технологии спасут мир. Всю свою сознательную жизнь я слышала обещания о том, что технологии изменят мир к лучшему. В сентябре 1991 г. я поступила в Гарвард на отделение вычислительных машин. Спустя несколько месяцев в ЦЕРН (лаборатории физики под патронажем Европейской организации по ядерным исследованиям) Тим Бернерс-Ли создал первый в мире веб-сайт. Когда я была

на втором курсе, мой сосед по комнате купил NeXTcube — тот же самый черный квадратный компьютер, который Барнерс-Ли использовал в качестве сервера в ЦЕРН. Было весело. Нам удалось подключить высокоскоростное соединение в нашей комнате, благодаря чему мы смогли проверять почту на компьютере стоимостью \$5000. Другой сосед, который был слишком молод, чтобы посещать бостонский гей-бар, с этого же компьютера зависал на онлайн досках объявлений, чтобы познакомиться с парнями. Тогда было легко поверить в то, что в будущем все будет происходить онлайн.

Юные идеалисты моего поколения с радостью решили, что создаваемый нами онлайн мир будет лучше и справедливее нынешнего. В 1960-х гг. наши родители верили, что можно построить лучший мир, порвав с обычной жизнью и живя в коммунах. Мы наблюдали за тем, как они бросали наркотики и осознали, что коммуны не привели к искомым ответам. Однако у нас впереди был совершенно новый, неизведанный мир «киберпространства», который предстояло создать. Связь между этими примерами далеко не метафорична. По словам Фреда Тёрнера, автора книги об истории цифрового утопизма «От контркультуры к киберкультуре»¹, на становление интернет-культуры того времени в известной степени повлияло движение Нового коммунизма 1960-х гг. В 1995 г. в специальном выпуске журнала *Time* под названием «Добро пожаловать в киберпространство» Стюарт Бранд, создатель Каталога всей Земли (Whole Earth Catalog), наметил связь между контркультурой и революцией персональных компьютеров в эссе «Мы всем обязаны хиппи»². Эпоха зарождения интернета была весьма кайфовой.

Будучи третьекурсницей, я могла создать веб-страницу, запустить веб-сервер и написать код на шести языках программирования. Это было вполне естественным для студента, специализирующегося в математике, компьютерах и программировании. И я была одной из шести женщин на этой специальности в университете, где училось 20 000 студентов. Я знала двух девушек — специалистов по теории вычислительных машин. Остальные три казались мифом. Я ощущала себя изгоем и прекрасно видела причины, заставлявшие женщин бросать науку, технологии, разработки, математику и уходить со STEM-специальностей*. Я также знала, как можно устранить эту «неисправность», однако ни я, ни другие женщины не имели для этого достаточной власти и влияния. Поэтому я перешла на другую специальность.

После окончания колледжа я стала работать программистом. В мои обязанности входило создание симулятора, который, подобно рою пчел, вооруженных винтовками, должен был набрасываться на вновь созданные программы и проверять их на прочность. Работа была хорошей, но не приносила радости. Я снова почувствовала, будто не было вокруг никого, кто бы выглядел, как я, говорил, как я, и интересовался теми же вещами, что и я. Поэтому я оттуда ушла и стала журналистом.

Переместимся на несколько лет вперед: я вернулась в компьютерную науку в качестве дата-журналиста. Журналистика данных — это практика поиска и рассказывания историй посредством цифр. Программирование

* Science Technology Engineering And Math (STEM) — аббревиатура, обозначающая технические специальности, в которых традиционно наблюдался гендерный перевес. — *Прим. пер.*

помогает проводить журналистские расследования. А еще я преподаю в университете, это мне подходит. И гендерный баланс там лучше.

Журналистов учат быть скептиками. Мы часто повторяем: «Если мама говорит, что любит тебя, это необходимо проверить». В течение многих лет я слышала фактически одни и те же обещания о светлом технологическом будущем, однако лично видела, как цифровой мир в точности повторял все нюансы неравенства в реальном мире. Например, процент женщин в пространстве технологических профессий никогда радикально не возрастал. Интернет превратился в новое публичное пространство, однако друзья и коллеги говорят, что именно онлайн они подвергаются харассменту больше, чем когда-либо. Мои подруги, пользовавшиеся сайтами и приложениями для знакомств, рассказывали об угрозах изнасилования и непристойных фотографиях, отправленных им. А тролли и боты превратили Twitter в какофонию голосов.

Тогда я начала серьезно задумываться над обещаниями, данными технокультурой, и стала замечать, как люди говорили о технологиях, не задумываясь о последствиях их внедрения. В конце концов, в основе всего, что мы делаем на компьютере, лежат математические действия, которые имеют свои фундаментальные ограничения, определяющие границы технологий. И, как мне кажется, мы приблизились к этим границам. Американцы достигли той точки развития, когда цифровые технологии с восторгом используются буквально везде — при приеме на работу, во время вождения, при оплате счетов и даже при выборе партнера, а способность критического осмысления новой реальности угасает на глазах.

Восторженные попытки применить компьютерные технологии в каждом аспекте нашей жизни привели к появлению невероятного количества недоработанных технологий. Они усложняют повседневную жизнь вместо того, чтобы делать ее проще. Так, например, поиск номера телефона нового знакомого или актуального адреса электронной почты занимает неожиданно большой объем времени. Проблема в том, что, как и во многих других случаях, вокруг слишком много новых технологий и недостаточно людей для управления ими. Мы перешли от протоколирования к компьютерным системам и при этом избавились от людей, способных содержать информацию в актуальном состоянии. И сегодня, поскольку никто не проверяет точность контактной информации в каждом институциональном справочнике, связаться с людьми оказывается гораздо сложнее. Как журналисту мне приходится общаться со множеством людей, которых я не знаю лично, и связываться с ними стало сложнее и дороже, чем раньше.

Есть поговорка «Когда у тебя в руках молоток, все кажется гвоздями». Компьютеры — это молотки. И пора наконец остановить слепую погоню за цифровым будущим и начать принимать более осознанные решения относительно того, когда и зачем использовать их.

Так появилась эта книга — некоторые размышления, помогающее понять, что могут и что не могут технологии, где технологические достижения пересекаются с человеческой природой. Эта грань похожа на обрыв, за которым нас ждет опасность.

В мире существует множество невероятных технологий: интернет-поисковики, устройства, распознающие вербальные команды, компьютеры, соревнующиеся

с людьми в играх вроде Jeopardy!* или го. Однако не стоит забывать о том, что как раз благодаря технологиям мы можем решать и определенные проблемы. В рамках моих университетских курсов я среди прочего учу одной фундаментальной вещи, а именно тому, что существуют ограничения. Подобно границам познаваемости в математике и естественных науках, имеются границы возможного применения технологий. Кроме того, существуют и пределы того, как нам *следует* использовать технологии. Смотря на мир через призму компьютерных наук или пытаюсь решить глобальные проблемы только при помощи технологий, мы рискуем совершить известные и ожидаемые ошибки, замедляющие прогресс и усиливающие существующее неравенство. Наша книга посвящена способам понимания внешних ограничений того, на что способны технологии. Осмысление этих рамок поможет нам совершать более качественный выбор и сформировать коллективный диалог относительно применения технологий и того, что следует делать, чтобы сделать мир настоящему лучше для всех.

К таким размышлениям меня привела журналистика. Я специализируюсь на разделе журналистики данных под названием *вычислительная журналистика*, ее также называют *анализом алгоритмической ответственности*. Алгоритм — это вычислительная процедура с известным результатом; он похож на рецепт, который при выполнении всех шагов приводит к созданию определенного «блюда». Иногда под анализом алгоритмической ответственности понимается процесс написания программы, при помощи

* В России выходит на канале НТВ под названием «Своя игра». — Прим. ред.

которой можно обнаружить алгоритмы, призванные принимать решения за пользователей. А порой подразумевается анализ не слишком качественных технологий или неверно интерпретированных данных, в результате которого мы обнаруживаем проблемную зону.

Одной из таких проблемных зон я собираюсь поделиться в этой книге, а именно *техношовинизмом*. Техношовинизм — это слепая вера в то, что технологии являются ответом на все вопросы. И, несмотря на то что цифровыми технологиями ученые и чиновники пользовались с 1950-х гг., а с 1980-х гг. компьютеры стали частью нашей повседневности, изощренные маркетинговые кампании все еще заставляют большинство людей верить в то, что цифровые технологии — это что-то совершенно новое и революционное. (Тем временем технологическая революция уже случилась, а цифровые технологии стали обыденностью.)

Техношовинизм нередко встречается вместе с созвучными ему идеями вроде меритократии в духе Айн Рэнд, технолибертарианских политических ценностей, торжества свободы высказывания до той степени, что бытование онлайн-харассмента не считается проблемой; с верой в то, что компьютеры более «объективны» и «беспристрастны», поскольку для них вопросы и ответы сведены к математическим уравнениям; и непоколебимой веры в то, что, если бы человечество больше полагалось на технологии и их правильное использование, социальные проблемы бы исчезли и мы создали утопическое общество на базе цифровых технологий. Никогда не было и не будет технологической инновации, способной избавить нас от необходимости сталкиваться с человеческой сущностью. Почему тогда люди упорно думают, будто где-то

за поворотом человечество обязательно найдет свое светлое технологическое будущее?

Я задумалась о техношовинизме после встречи со знакомым специалистом по обработке данных, которому на тот момент было около 20 лет. Я тогда вскользь упомянула о том, что в школах Филадельфии недостаточно книг.

— Почему попросту не пользоваться ноутбуками, планшетами и электронными книгами? — спросил знакомый. — Технологии ведь делают все быстрее, дешевле и лучше, не так ли?

Он получил нагоняй (который и вы получите в следующей главе). И тем не менее его предположение засело в моей голове. Мой друг был уверен, что технологии решали все проблемы. Я же считала, что они годятся только в качестве инструмента для решения конкретной задачи.

За последние два десятка лет странным образом сформировалось представление о том, что компьютер всегда все делает правильно, а человек — нет. Мы говорим, что «компьютеры лучше, потому что они объективнее людей». Компьютеры стали вездесущи и настолько проникли в нашу жизнь, что, когда барахлит машина, мы тут же виним себя вместо того, чтобы подумать о возможных ошибках в тысячах строках кода, составляющего обычную программу. Как и любой разработчик, скажу, что на самом деле проблема чаще всего скрыта в компьютере. В частности, это может быть недоработанный и плохо протестированный код, дешевое аппаратное обеспечение или банальное непонимание разработчиками практики использования софта конечным потребителем.

Если вы согласны с моим другом-исследователем, то, вероятно, к последней фразе вы относитесь скептически.

Может, вы тот самый человек, который обожает смартфоны, или всю вашу жизнь вам говорили, что компьютеры — дорога в будущее? Я вас прекрасно понимаю, мне тоже так говорили. Однако я прошу вас вместе со мной прочитать еще несколько историй о людях, которые создают технологии, и затем критически поразмыслить и о технологиях, и об их создателях. Эта книга — не инструкция и не учебник, она представляет собой сборник историй, объединенных одной идеей. Я выбрала несколько авантюрных историй из области программирования, каждая из которых позволяет понять нечто фундаментальное о технологиях и современной технокультуре. Все эти проекты соединяются в цепочку аргументов против техношовинизма. Кроме того, я расскажу, как функционируют компьютерные технологии, и покажу, как они служат потребностям человека.

Первые четыре главы посвящены базовым принципам работы компьютеров и основам концепции программирования. И если вам уже понятно, как аппаратное оснащение и софт работают вместе, или если вы знаете, как писать код, то стоит быстро пробежаться по главам 1–3 и перейти сразу к главе 4, посвященной данным. Первые три главы важны, поскольку дают представление о том, что любой искусственный интеллект (ИИ) работает на основе одних и тех же элементов — кода, данных, двоичной логики и электрических импульсов. И важно понять, что в ИИ является реальным, а что воображаемым. Искусственные суперинтеллекты вроде тех, что мы видим в телешоу «В поле зрения» или «Звездный путь», — выдуманные. Да, конечно, их интересно представлять себе вживую, и они вдохновляют многих людей на размышления о способности роботов завоевать мир

и тому подобном, но они не настоящие. Эта книга помогает прорваться к реальным математическим, когнитивным и вычислительным концептам, бытующим в современной научной дисциплине ИИ: представлению знаний и построению логических рассуждений, логике, машинному обучению, обработке естественных языков, поиску, планированию, механике и этике.

В рамках первого вычислительного приключения (глава 5) я анализирую причины, по которым после двух десятилетий реформирования образования школы до сих пор неспособны подготовить учеников к сдаче стандартизированных тестов. И это не вина учеников или учителей. Проблема гораздо масштабнее: компании, разрабатывающие государственные и локальные школьные тесты, также издают учебники, в которых есть ответы на тесты. Однако далеко не все школы могут себе позволить такие учебники.

Я обнаружила эту щекотливую ситуацию, пока писала код ИИ для своих журналистских расследований. *Associated Press* использует ботов для написания новостных заметок о бизнесе и спорте, так что роботы-репортеры — явление нередкое для последних лет. Моя программа не находилась непосредственно в работе-репортере (в этом не было нужды, но я и не отрицаю такую возможность), она также не писала истории (по тем же причинам). Напротив, эта программа была принципиально новым способом применения старого доброго искусственного интеллекта, она помогала обнаружить интересные моменты. Одним из наиболее удивительных открытий, сделанных в ходе вычислительного расследования, стало то, что даже в нашем высокотехнологичном мире простейшее решение — книга в руках ребенка — оказалось весьма эффективным.

Это заставило меня задуматься о том, почему мы тратим так много денег на внедрение технологий в классах, когда у нас уже есть дешевое и эффективное решение, которое неплохо работает.

В главе 6 нас ждет насыщенный ретроспективный обзор истории компьютеров. Особое внимание будет отдано Марвину Минскому — человеку, известному в качестве отца искусственного интеллекта, — и той огромной роли, которую контркультура 1960-х гг. сыграла в формировании мнений об интернете, существующем в 2017 г., в котором была написана эта книга. Я хочу показать, как мечты и цели определенных людей сформировали научное знание, культуру, деловую риторику и даже правовые рамки современных технологий посредством череды взвешенных решений. Например, причина, по которой интернет не поделен на государственные территории, заключается в том, что создатели этой технологии стремились построить новый мир за пределами государств — подобно тому, что они (безуспешно) пытались построить в коммунах.

Размышляя о технологиях, необходимо также помнить о другом краеугольном камне массовой культуры — о Голливуде. Большая часть представлений о технологиях сформирована благодаря фильмам, телепередачам и книгам. (Помните робота из моего детства?) Говоря об ИИ, нужно различать сильный и слабый искусственный интеллект. Сильный — это голливудская версия. Как раз благодаря такому ИИ оживает робот-дворецкий, который теоретически может обрести сознание и захватить государство, что, в свою очередь, может привести к появлению настоящего Арнольда Шварценеггера в качестве Терминатора и иным не слишком приятным последствиям.

Большинство исследователей в области вычислительной техники читают научно-фантастическую литературу и смотрят кино, поэтому всегда рады обсудить гипотетические возможности сильного искусственного интеллекта.

В 1990-х гг. исследователи поставили крест на сильном ИИ³. Сегодня его называют «старым добрым искусственным интеллектом». Слабый ИИ — настоящий. Он опирается исключительно на вычислительные методы и не настолько увлекателен, как его более старый собрат, но удивительно хорошо справляется с разного рода задачами. Хотя, конечно, существует знаковая лингвистическая путаница. Машинное обучение (МО) — популярная ныне форма искусственного интеллекта — не является сильным ИИ. Это слабый ИИ, хотя название действительно может ввести в заблуждение. Даже для меня фраза «машинное обучение» ассоциируется с каким-то разумным существом внутри компьютера.

Важное различие заключается в следующем: сильный ИИ — это то, чего мы желаем, на что надеемся и что представляем себе (без учета злобных роботических повелителей эпохи расцвета научной фантастики). Слабый ИИ — то, что у нас реально есть. Такова разница между мечтой и реальностью.

Затем, в главе 7, я представлю принцип работы МО и покажу, как создать такую структуру, которая способна предсказать, кто из пассажиров «Титаника» выжил бы в известном крушении. Понимание принципа работы МО позволит разобраться в примере из главы 8, где я окажусь за рулем автономной машины и расскажу, почему беспилотный школьный автобус обязательно попадет в аварию. Впервые в жизни я села в беспилотный автомобиль в 2007 г., тогда компьютерный «водитель» чуть

не убил меня на стоянке фирмы «Боинг». С тех пор технологии прошли долгий путь, однако принципиально так и не стали работать так же хорошо, как человеческий мозг. Так что в ближайшее время киборгизированного будущего не наступит. Я также обращаю внимание на то, как люди представляют технологии, заменяющие человека, и анализирую, почему так сложно принять тот факт, что технологии не настолько эффективны, как мы того хотим.

Глава 9 станет плацдармом для размышлений о том, почему *популярно* не значит *хорошо* и почему это заблуждение — подкрепляемое машинным обучением — по-настоящему опасно. Главы 10 и 11 — очередные программистские приключения, где я создаю пицце-расчетную компанию на междугородном хакатон-автобусе (популярно, но не очень хорошо) и пытаюсь исправить финансовую систему США к президентским выборам 2016 г. (хорошо, но не слишком популярно). В обоих случаях я создаю софт, который работает не так, как изначально задумывалось, и его крах весьма поучителен.

С помощью этой книги я хочу воодушевить людей. Я хочу, чтобы они поняли принципы работы компьютера и перестали бояться программ. Мы все когда-то были в такой ситуации. Все чувствовали беспомощность и расстраивались перед, казалось бы, простой задачей, которая по факту становится невыполнимой из-за технологического интерфейса. Даже мои студенты, которых порой называют «цифровое поколение», иногда считают, что цифровой мир сбивает с толку, пугает и недостаточно проработан.

Полагаясь на технологии при решении сложных социальных задач, мы, соответственно, полностью рассчитываем

на искусственную неразумность. Фактически именно компьютер, а не человек, и есть эта искусственная неразумность. Компьютеру все равно, что делает он или пользователь. Он всего лишь выполняет команды так хорошо, как только может, затем ждет следующую команду. У него нет сознания, нет души.

Человечество разумно. В то же время умные и доброжелательные люди действуют как техношвинисты просто потому, что не замечают негативных последствий принятия решений компьютерами либо они крайне привержены идее использования компьютеров повсеместно (и даже при выполнении задач, к которым компьютеры категорически не приспособлены).

Мне кажется, мы способны на большее. Как только мы поймем, как действительно работают компьютеры, мы сможем предъявлять более высокие требования к качеству технологий. Вместо того чтобы мириться с системами, которые только лишь обещают улучшения, а на самом деле все усложняют, мы можем требовать такие системы, которые в действительности все делают дешевле, быстрее и лучше. Мы можем научиться принимать качественно иные решения о следствиях развития технологий так, чтобы неосознанно не навредить сложным социальным системам. И мы можем почувствовать в себе силы сказать нет необязательным технологиям, чтобы начать жить более качественной жизнью и наслаждаться тем, как технологии обогащают наш мир.

2

Hello, world!

Чтобы понять, что компьютеры *не могут*, прежде необходимо разобраться, как они работают и с какими задачами справляются. Для этого напишем простую компьютерную программу. Начиная изучать новый язык программирования, специалист обычно пишет программу «Hello, world!». Не важно, изучаете ли вы программирование в учебном лагере, в Стэнфорде, в университете или онлайн, — вы, скорее всего, тоже ее напишете. «Hello, world!» — это отсылка к первой программе в легендарной книге Брайана Кернигана и Дениса Ричи «Язык программирования С»* (The C Programming Language), где читателю предлагается написать программу (с помощью языка С, разумеется), выводящую на экран эту фразу. Керниган и Ричи работали в лаборатории Bell — исследовательском центре, статус которого в индустрии сравним с Hershey в мире шоколада (AT&T Bell Labs были очень добры, пригласив меня на несколько лет на работу). Именно здесь зарождалось множество инноваций, в том числе лазер, микроволновка и Unix (Ричи помогал и в разработке Unix, и в разработке

* Керниган Б. Ритчи Д. Язык программирования С. — М.: Вильямс, 2009.

языка C). Язык назвали C потому, что до этого команда лаборатории уже изобрела язык под названием B. Все еще популярный C++ и его двоюродный брат C# — потомки языка C.

Я люблю традиции, поэтому давайте поддержим одну из них и напишем «Hello, world!». Пожалуйста, возьмите листок бумаги, ручку и напишите «Hello, world!».

Мои поздравления! Это было просто.

А «за кадром» было чуть сложнее. Вы сфокусировали внимание, взяли необходимые инструменты, чтобы реализовать свое намерение, скомандовали руке писать буквы, а также использовали другую руку или иные части вашего тела, чтобы крепко зафиксировать бумагу во время письма, — так выглядела механика процесса. Вы скомандовали своему телу выполнять пошаговый алгоритм для достижения конкретной цели.

Осталось заставить компьютер сделать то же.

Откройте текстовый редактор — это может быть все что угодно: Microsoft Word, Notes, Pages или OpenOffice — и создайте новый документ. В этом документе наберите «Hello, world!». Можете распечатать, если хотите.

Опять мои поздравления! Вы использовали новый инструмент для выполнения той же задачи: намерение, механика и т.д. Вы на волне успеха.

Следующее испытание заключается в том, чтобы заставить компьютер вывести на экран фразу «Hello, world!» немного иным образом. Мы напишем программу, которая сама выведет ее. Для этого мы используем язык программирования Python, установленный на всех компьютерах Mac. (Если вы не используете Mac, то процесс может выглядеть слегка иначе, вам нужно будет найти инструкции в интернете.) Открываем «Приложения» и видим среди

прочих программу «Терминал» (см. рис. 2.1). Открываем ее.

И снова примите мои поздравления! Вы только что улучшили свои навыки владения компьютером. Сейчас вы приблизились к компьютерному «железу».

Под «железом» имеется в виду аппаратное обеспечение компьютера: чипы, транзисторы, провода и т.п. Это то, что составляет физический облик компьютера. Запуская систему, вы видите приятный пользовательский интерфейс, который как раз и обеспечивает доступ к железу компьютера. Мы воспользуемся терминалом, чтобы написать программу на языке Python, которая как раз выведет надпись «Hello, world!» на экран.

В терминале вы увидите мигающий курсор. Он показывает *командную строку*. Компьютер интерпретирует —



Рис. 2.1. Программа «Терминал» в папке «Утилиты»

достаточно буквально — все, что вы в ней напишете. В общем, когда вы нажмете Enter, компьютер попытается выполнить заданную команду. Итак, попробуем написать следующее:

```
python
```

Далее вы увидите нечто подобное:

```
Python 3.5.0 (default, Sep 22 2015, 12:32:59)
[GCC 4.2.1 Compatible Apple LLVM 7.0.0 (clang-700.0.72)]
on darwin
Type "help," "copyright," "credits" or "license" for
more information.
>>>
```

Символ тройной шляпки (>>>) показывает, что вы вошли в режим интерпретации языка Python и одновременно вышли из базового интерпретатора команд. Последний использует особый тип языка программирования, *язык оболочки*. Соответственно, интерпретатор Python использует язык Python. В программировании, как и в естественных языках, существует множество диалектов.

Напишите следующее и нажмите Enter.

```
print («Hello, world!»)
```

Отлично! Вы только что написали компьютерную программу. И как ощущения?

Мы только что сделали одну и ту же вещь тремя разными способами. Один из них, пожалуй, был более приятным, чем прочие. Другой — быстрее и проще. Опираясь на полученный опыт, вы можете решить, какой из способов проще, а какой — быстрее. Однако принципиальный момент заключается в том, что *ни один из них не лучше других*.

Поэтому, когда мы говорим, что делать что-либо при помощи технологий лучше, это сродни утверждению, будто написать «Hello, world!» лучше на языке Python, а не на бумаге. Не существует какой-либо принципиальной ценности в каждом из этих способов, однако их применение зависит от индивидуального опыта и последствий выбора в реальном мире. В случае с фразой «Hello, world!» ставки невысоки.

Большинство программ сложнее нашего примера, однако понимание принципа позволяет масштабировать его и применять в более комплексных случаях. Каждая программа — от сложнейших научных вычислений до новой социальной сети — пишется людьми. И каждый из них начинал программировать с «Hello, world!». Их путь к созданию сложного, изощренного программного обеспечения начинался с программ, построенных из простых блоков вроде нашего «Hello, world!». Компьютерные программы — не магия. Они созданы человеком.

Допустим, я хочу написать программу, которая бы выводила «Hello, world!» на экран десять раз. В этом случае я могу 10 раз написать команду:

```
print ("Hello, world!")  
print ("Hello, world!")
```

Ой, нет. Я уже устала, забудем об этом. Нажимать Ctrl + P на клавиатуре еще восемь раз — слишком много нажатий кнопок. (Чтобы думать, как программист, нужно научиться быть ленивым.) Большинство программистов уверены, что печатать на клавиатуре скучно и утомительно, поэтому они стараются делать это как можно реже. Вместо перепечатывания, копирования и вставки единственной строки кода я введу оператор цикла, чтобы компьютер повторил команду 10 раз.

```
x=1
while x<=10:
    print ("Hello, world! \ n")
    x+=1
```

Что ж, это было весело! Теперь компьютер будет делать всю работу за меня. Подождите! А что сейчас произошло?

Я задала значение x , равное 1, и добавила оператора цикла WHILE, который и будет повторять программу до тех пор, пока она не достигнет значения $x > 10$. В первом цикле $x = 1$. Программа выводит текст, затем следует разрыв строки или знак конца строки, обозначенный `\n` (обратный слеш n). Обратный слеш — важный знак в языке Python. Интерпретатор языка «знает», что, как только в коде появляется этот знак, после него должно произойти что-то особенное. В нашем случае я прошу компьютер начать с новой строки. Это было бы мукой — начинать каждый раз с нуля и программировать каждый тупой кусок металла для выполнения одних и тех же базовых функций, таких как чтение текста и преобразование его в двоичную форму или выполнение определенных задач в соответствии с соглашениями синтаксиса выбранного нами языка программирования. Мы ничего и никогда не закончили бы такими темпами! Более того, в каждом компьютере присутствуют как изначально встроенные функции, так и возможность добавлять новые. Я использую слово «знает», потому что оно удобное, но, пожалуйста, не забывайте, что компьютеры не «знают» подобно тому, как «знают» сознательные существа. Внутри компьютера нет никакого сознания, есть только задачи, выполняемые бесшумно, быстро и хорошо.