

My, programiści

Kronika koderów od Ady do AI

Tytuł oryginału: We, Programmers: A Chronicle of Coders from Ada to AI (Robert C. Martin Series)

Tłumaczenie: Grzegorz Werner

ISBN: 978-83-289-2650-9

Authorized translation from the English language edition, entitled WE, PROGRAMMERS: A CHRONICLE OF CODERS FROM ADA TO AI, 1st Edition 9780135344262 by Robert C. Martin published by Pearson Education, Inc, publishing as Addison-Wesley Professional.

Copyright © 2025 Pearson Education.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

POLISH language edition published by HELION S.A., Copyright © 2025.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

helion.pl/user/opinie/myprkr

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 230 98 63

e-mail: *helion@helion.pl*

WWW: *helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

SPIIS TREŚCI

Słowo wstępne	15
Przedmowa	17
Linia czasu	20
O książce	23
Podziękowania	24
O autorze	25
 CZĘŚĆ I	
Przygotowanie sceny	27
 Rozdział 1.	
Kim jesteśmy?	29
Dlaczego tu jesteśmy?	31
 CZĘŚĆ II	
Giganci	35
 Rozdział 2.	
Babbage — pierwszy inżynier komputerowy	37
Człowiek	37
Tablice	39
Tworzenie tablic	39
Różnice skończone	41

	Wizja Babbage’a	45
	Maszyna różnicowa	46
	Notacja mechaniczna	47
	Imprezowe sztuczki	48
	Upadek maszyny	49
	Argument technologiczny	50
	Maszyna analityczna	50
	Symbole	53
	Ada, hrabina Lovelace	53
	Pierwszy programista?	57
	Tylko dobrzy umierają młodo	57
	Mieszane zakończenie	57
	Realizacja maszyny różnicowej 2	58
	Podsumowanie	59
	Źródła	59
Rozdział 3.	Hilbert, Turing i von Neumann — pierwsi architekci komputerowi	61
	David Hilbert	62
	Gödel	64
	Burzowe chmury	67
	John von Neumann	67
	Alan Turing	70
	Architektura Turinga-von Neumanna	73
	Maszyna Turinga	73
	Podróż von Neumanna	77
	Źródła	85
Rozdział 4.	Grace Hopper — pierwszy inżynier oprogramowania	86
	Wojna i lato 1944 roku	87
	Dyscyplina (1944 – 1945)	91
	Podprogramy (1944 – 1946)	95
	Symposium (1947)	96
	UNIVAC (1949 – 1951)	98
	Sortowanie i początki kompilatorów	103
	Alkohol (około 1949)	104
	Kompilatory (1951 – 1952)	105
	Kompilatory typu A	106
	Języki (1953 – 1956)	108
	COBOL (1955 – 1960)	110

	Moja refleksja na temat COBOL-a	113
	Niewątpliwy sukces	113
	Źródła	114
Rozdział 5.	John Backus i pierwszy język wysokopoziomowy	116
	John Backus, człowiek	116
	Hipnotyzujące kolorowe światelka	118
	Speedcoding i 701	120
	Potrzeba prędkości	123
	Podział pracy	126
	Moja refleksja na temat FORTRAN-u	128
	ALGOL i cała reszta	129
	Źródła	131
Rozdział 6.	Edsger Dijkstra — pierwszy informatyk	132
	Człowiek	132
	ARRA (1952 – 1955)	134
	ARMAC (1955 – 1958)	138
	Algorytm Dijkstry — najkrótsza ścieżka	139
	ALGOL i X1 (1958 – 1962)	140
	Gęstniejący mrok (1962)	144
	Wzrost znaczenia nauki (1963 – 1967)	145
	Nauka	145
	Semaforey	146
	Struktura	147
	Dowód	147
	Matematyka (1968)	148
	Programowanie strukturalne (1968)	151
	Argument Dijkstry	152
	Źródła	154
Rozdział 7.	Nygaard i Dahl — pierwszy język obiektowy	155
	Kristen Nygaard	155
	Ole-Johan Dahl	157
	Simula i obiektowość	158
	Simula I	162
	Źródła	168

Rozdział 8.	John Kemeny i BASIC, pierwszy język „dla każdego”	169
	John Kemeny, człowiek	169
	Thomas Kurtz, człowiek	171
	Rewolucyjna idea	172
	Niemożliwe	173
	BASIC	174
	Podział czasu	175
	Komputerowe dzieciaki	176
	Ucieczka	177
	Ślepy prorok	177
	Symbioza?	178
	Przepowiednie	178
	Jakby w zwierciadle, niejasno	182
	Źródła	182
Rozdział 9.	Judith Allen	184
	ECP-18	185
	Judith Schultz	186
	Wspaniała kariera	189
	Źródła	190
Rozdział 10.	Thompson, Ritchie i Kernighan	191
	Ken Thompson	191
	Dennis Ritchie	193
	Brian Kernighan	197
	Multics	199
	PDP-7 i podróże kosmiczne	200
	Unix	203
	PDP-11	207
	C	208
	K&R	211
	Wykręcanie rąk	213
	Narzędzia programowe	214
	Podsumowanie	214
	Źródła	215

CZĘŚĆ III	Wygięcie krzywej	217
Rozdział 11.	Lata sześćdziesiąte	219
	ECP-18	223
	Co robią ojcowie?	225
Rozdział 12.	Lata siedemdziesiąte	226
	1969	226
	1970	231
	1973	233
	1974	236
	1976	241
	Kontrola kodu źródłowego	243
	1978	244
	1979	246
	Źródła	247
Rozdział 13.	Lata osiemdziesiąte	248
	1980	248
	Administrator systemu	250
	pCCU	250
	1981	251
	DLU/DRU	251
	Apple II	253
	Nowe produkty	253
	1982	254
	Xerox Star	255
	1983	256
	Wewnątrz Macintosha	257
	BBS-y	257
	C w Teradyne	257
	1984 – 1986: VRS	257
	Wojny rdzeniowe	258
	1986	259
	Craft Dispatch System (CDS)	259
	Dane z etykietami pól	260
	Automaty skończone	260
	Programowanie obiektowe	261
	1987 – 1988: Wielka Brytania	262
	Źródła	263

Rozdział 14.	Lata dziewięćdziesiąte	264
	1989 – 1992: Clear Communications	264
	Usenet	265
	Wujek Bob	265
	1992: „The C++ Report”	267
	1993: Rational Inc.	267
	1994: ETS	268
	Felieton w „The C++ Report”	270
	Wzorce	270
	1995 – 1996: pierwsza książka, konferencje, kursy i Object Mentor Inc.	271
	Zasady	272
	1997 – 1999: „The C++ Report”, UML i dotcomy	273
	Druga książka: zasady projektowe	273
	1999 – 2000: programowanie ekstremalne	274
	Źródła	276
 Rozdział 15.	 Milenium	 277
	2000: liderzy XP	277
	2001: Agile i krach	278
	2002 – 2008: wędrówka po pustyni	279
	Czysty kod	280
	2009: SICP i kluczkowanie chromatyczne	280
	Wideo	281
	cleancoders.com	282
	2010 – 2023: wideo, rzemiosło i profesjonalizm	282
	Zmierzch Agile	283
	Więcej książek	283
	Pandemia COVID-19	284
	2023: plateau	284
	Źródła	285
 CZĘŚĆ IV	 Przyszłość	 287
 Rozdział 16.	 Języki	 289
	Typy	291
	Lisp	292
 Rozdział 17.	 Sztuczna inteligencja	 294
	Ludzki mózg	294
	Sieci neuronowe	296

	Budowanie sieci neuronowych to nie programowanie	298
	Duże modele językowe	298
	Destabilizujący wpływ dużych modeli X-owych	304
Rozdział 18.	Sprzęt	306
	Prawo Moore’a	307
	Rdzenie	308
	Chmura	308
	Plateau	308
	Komputery kwantowe	308
Rozdział 19.	WWW	310
Rozdział 20.	Programowanie	314
	Analogia lotnicza	315
	Zasady	315
	Metody	316
	Dyscypliny	316
	Etyka	316
	Źródła	317
Posłowie		319
	Refleksje na temat książki	319
	Osobiste anegdoty i opowieści	320
	Refleksje na temat książki	326
	Punkt widzenia autora posłowania	326
	Omówienie przyszłych trendów	327
	Wezwanie do działania albo końcowe wnioski	329
	Źródła	329
Glosariusz		330
Spis postaci drugoplanowych		350

BABBAGE — 2 PIERWSZY INŻYNIER KOMPUTEROWY

Wśród programistów i miłośników komputerów powszechnie wiadomo, że Charles Babbage był ojcem komputera uniwersalnego, a Ada King Lovelace była pierwszym programistą. Opowiedziano o tym wiele barwnych historii, zarówno fikcyjnych, jak i paradokumentalnych. Ale, jak zawsze, prawda jest o wiele ciekawsza niż legenda.

Człowiek

Charles Babbage urodził się w Walworth w Surrey 26 grudnia 1791 roku. Jako syn zamożnego bankiera, należał do wyższych sfer brytyjskiego społeczeństwa początku XIX wieku i odziedziczył pokaźny majątek¹, który pozwolił mu na oddawanie się zainteresowaniom — a miał ich mnóstwo.

W ciągu swojego życia Babbage napisał 6 książek i 86 prac naukowych na takie tematy, jak matematyka, szachy, wytrychy, podatki, ubezpieczenia na życie, geologia, polityka, filozofia, elektryczność i magnetyzm, instrumentacja, statystyka, kolejnictwo, obrabiarki, ekonomia polityczna, aparaty do nurkowania, łodzie podwodne, nawigacja, podróże, filologia, kryptoanaliza, sztuki przemysłowe, astronomia i archeologia — żeby wymienić tylko niektóre.

¹ Majątek wart 100 000 funtów, który dał mu niezależność finansową i pozwolił zapewnić rodzinie dostatek, a jednocześnie finansować własne badania naukowe.

Ale przede wszystkim Babbage był majsterkowiczem — wynalazcą najróżniejszych mechanizmów. Wynalazł system ogrzewania nadmuchowego, oftalmoskop, system dostarczania poczty za pomocą kabla, maszynę do gry w kółko i krzyżyk oraz wiele innych urządzeń. Jednak w całym tym majsterkowaniu nie udało mu się stworzyć niczego, co przyniosłoby mu zysk. W większości przypadków jego wynalazki były rysunkami, które nigdy nie opuszczały stołu kreślarskiego.

Babbage był również bardzo towarzyski. Był gawędziarzem i artystą *par excellence*. Organizował u siebie kolacje, a jego obecność na przyjęciach u innych osób była bardzo pożądana. W pewnym momencie, w 1843 roku, jego kalendarz towarzyski obejmował 13 zaproszeń na każdy dzień miesiąca, łącznie z niedzielami².

Do jego znajomych należeli między innymi Charles Dickens, Charles Darwin, Charles Lyell, Charles Wheatstone (wielu Charlesów), George Boole, George Biddell Airy, Augustus De Morgan, Alexander von Humboldt, Peter Mark Roget, John Herschel i Michael Faraday.

Zdobył również wiele zaszczytów. Został wybrany członkiem Royal Society w 1816 roku. W 1824 roku otrzymał pierwszy złoty medal Astronomical Society za wynalezienie maszyny obliczeniowej, o której wkrótce będziemy mówić. W 1828 roku objął katedrę Lucasian Chair of Mathematics w Cambridge³, którą kierował do 1839 roku.

Trzeba powiedzieć, że był popularnym i dobrze ustosunkowanym facetem.

Pomimo popularności w towarzystwie i uznania współczesnych Babbage nie odniósł wielkich sukcesów. Większość jego przedsięwzięć spełzała na niczym. Nie było też przyjemnie z nim pracować. Uważano go za kapryśnego i kłótliwego, skłonnego do nieprzyjemnych i samolubnych wybuchów — porywczego geniusza⁴.

Miał zwyczaj publikowania listów, w których krytykował osoby sprawujące władzę — te same, które prosił o wsparcie finansowe swoich projektów. Powiedzmy, że dystygnowana dyskrecja nie była jedną z jego cnót.

Na koniec nawet premier Sir Robert Peel zapytał: „Co zrobimy, aby pozbyć się pana Babbage’a i jego maszyny liczącej?”.

² Swade, s. 173.

³ Na katedrze Lucasa zasiadało wiele innych ważnych postaci, w tym Isaac Newton (1669 – 1702), Paul Dirac (1932 – 1969), Stephen Hawking (1979 – 2009) oraz Data (gdzieś po 2395 r.).

⁴ *Irascible Genius* (Porywczy geniusz) to tytuł książki o Babbage’u napisanej przez Mabothe Moseleya w 1964 r. Zobacz: <https://archive.org/details/irasciblegeniusl00mose/mode/2up>.

Tablice

Historia Charlesa Babbage’a jako programisty zaczyna się latem 1821 roku. Wówczas Babbage i jego wieloletni przyjaciel, John Herschel, wspólnie sprawdzają tablice matematyczne dla Astronomical Society. Istnieją dwa zestawy tablic stworzone przez dwa niezależne zespoły. Jeśli oba zespoły wykonały wszystkie obliczenia poprawnie, obie tablice powinny być identyczne. Babbage i Herschel porównują je, aby znaleźć różnice. Tablice zawierają tysiące liczb, z których każda składa się z co najmniej kilkunastu cyfr. Obaj mężczyźni poświęcają długie godziny żmudnej, mozolnej, wymagającej skupienia pracy, recytując sobie nawzajem liczby, by zweryfikować, czy są identyczne. Za każdym razem gdy w zapisanej liczbie pojawi się błąd lub zostanie ona źle odczytana, muszą zatrzymać się, sprawdzić liczbę ponownie i albo usunąć niezgodność, albo opatrzyć ją adnotacją. Jest to ogłupiająca, frustrująca i wyczerpująca praca. Na koniec Babbage wykrzykuje: „Boże, chciałbym, żeby te obliczenia wykonała maszyna parowa!”⁵.

I w taki sposób narodził się Babbage programista. Nim minął rok, zapłacił rzemieślnikom za zbudowanie części, a następnie złożył mały działający model maszyny, która potrafiła liczyć. Był to mały prototyp jego wizji o wiele większej maszyny. Maszyny, która miała wykonywać niewdzięczną pracę polegającą na obliczaniu tablic matematycznych.

Tworzenie tablic

Tablice matematyczne były potrzebne wszystkim. Potrzebowali ich matematycy. Potrzebowali ich nawigatorzy. Potrzebowali ich astronomowie, inżynierowie i geodeci. Potrzebowali tablic logarytmicznych, tablic trygonometrycznych, tablic balistycznych, tablic pływów. Lista ciągnęła się w nieskończoność. Co więcej, potrzebowali zarówno dokładności, jak i precyzji. Każdy wpis w tablicy musiał być dokładny i precyzyjny do kilku miejsc po przecinku.

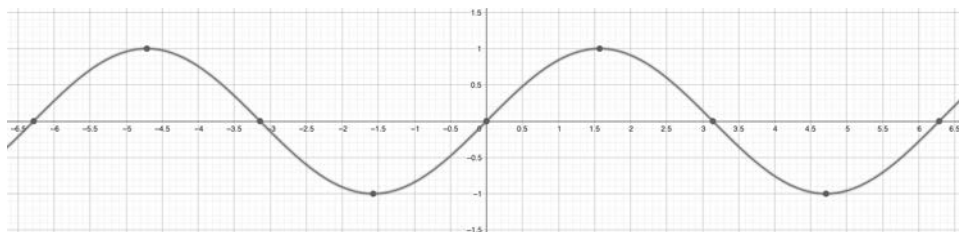
Jak powstają takie tablice? Jak obliczyć dziesiątki tysięcy logarytmów do sześcianu, ośmiu lub dziesięciu miejsc po przecinku? Jak wyznaczyć sinusy, cosinusy i tangensy kąta sekunda po sekundzie kątowej z taką precyzją? Na pierwszy rzut oka problem wydaje się nie do rozwiązania.

Ale pomimo pewnych dowodów na to, że jest inaczej, ludzie to sprytne bestie. Okazuje się, że jest na to sposób.

Pierwszą częścią sztuczki jest sprowadzenie tego, co transcendentne, z powrotem na Ziemię. Logarytmy, sinusy i cosinusy są funkcjami transcendentalnymi, co oznacza, że nie można ich obliczyć za pomocą wielomianu o współczynnikach wymiernych. *Można* je jednak *przybliżyć* za pomocą takiego wielomianu.

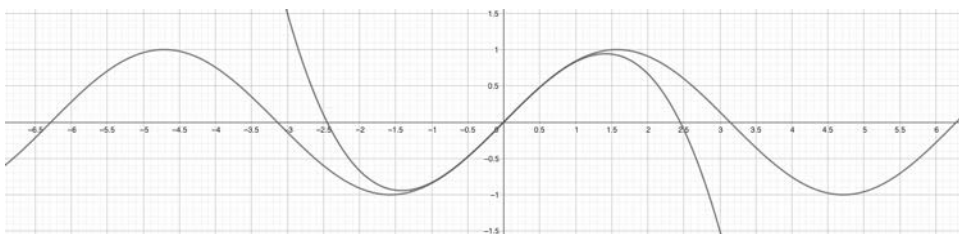
⁵ Swade, s. 10.

Rozważ falę sinusoidalną na kartezjańskiej płaszczyźnie współrzędnych. Faluje ona na osi y w zakresie od -1 do 1 z okresem 2π na osi x (rysunek 2.1).

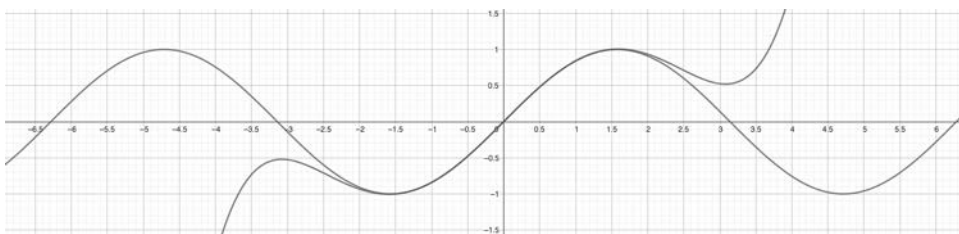


Rysunek 2.1. Sinusoida

Teraz rozważ wykres funkcji $y = -0,1666x^3 + x$ nałożony na tę sinusoidę (rysunek 2.2).

Rysunek 2.2. Funkcja $y = -0,1666x^3 + x$ nałożona na sinusoidę

Przybliżenie jest dość dobre, kiedy x jest bliskie zeru. Ale da się zrobić to lepiej. Rozważ funkcję $y = 0,00833x^5 - 0,1666x^3 + x$ (rysunek 2.3).

Rysunek 2.3. Funkcja $y = 0,00833x^5 - 0,1666x^3 + x$ nałożona na sinusoidę

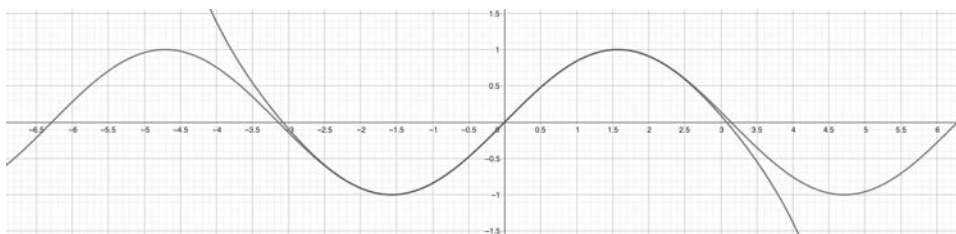
Oho! Jesteśmy coraz bliżej! Ale da się to zrobić jeszcze lepiej. Rozważ poniższą funkcję:

$$y = -0,0001984x^7 + 0,00833x^5 - 0,1666x^3 + x \quad (\text{rysunek 2.4}).$$

Wow! Pomiędzy $-\pi/2$ a $\pi/2$ jesteśmy naprawdę niedaleko. W rzeczywistości dla $-\pi/2$ wartość wielomianu wynosi $-1,00007$. To dość blisko czterech dziesiętnych cyfr precyzji.

Oczywiście aby uzyskać wszystkie te interesujące współczynniki, używam prostego rozwinięcia Taylora:

$$\sin(x) = x - x^3/3! + x^5/5! - x^7/7! + \dots$$



Rysunek 2.4. Funkcja $y = -0,0001984x^7 + 0,00833x^5 - 0,1666x^3 + x$ nałożona na sinusoidę

OK, skoro udało nam się wydobyć sinus z królestwa transcendencji, jak obliczymy wszystkie te paskudne wielomiany bez popadania we frustrację i szaleństwo?

No bo przypuśćmy, że chcemy zbudować tablicę sinusów dla każdej sekundy kątowej w zakresie od 0 do $\pi/2$. W zakresie tym są 324 000 sekund kątowych, a wartość jednej sekundy to 0,000004848136811 radiana. Czy naprawdę masz ochotę podnosić taką liczbę do trzeciej, piątej i siódmej potęgi, dzielić wyniki przez 6, 120 i 5040, a potem dodawać je i odejmować — 340 000 razy?

Na szczęście jest lepszy sposób: metoda różnic skończonych.

Różnice skończone

Wyobraź sobie prosty wielomian, taki jak $f(x) = x^2 + 3x - 2$. Zobaczmy, jakie przyjmuje wartości dla x od 1 do 5:

x	$x^2 + 3x - 2$
1	2
2	8
3	16
4	26
5	38

Teraz obliczmy pierwszą różnicę między tymi wartościami.

x	$x^2 + 3x - 2$	d1
1	2	
2	8	6
3	16	8
4	26	10
5	38	12

A teraz drugą różnicę:

x	x^2+3x-2	d1	d2
1	2		
2	8	6	2
3	16	8	2
4	26	10	2
5	38	12	2

Aha! Drugą różnicą jest stała 2. Dla dowolnego wielomianu stopnia n n -ta różnica będzie stała.

Jaka jest zatem wartość naszej funkcji, jeśli x jest równe 6? Zanim zaczniesz obliczać wielomian, zauważ, że pierwsza różnica powinna wynosić 14, ponieważ po prostu dodajemy drugą różnicę (2) do pierwszej (12). Ale teraz możemy dodać nową pierwszą różnicę do 38, aby otrzymać 52, co jest poprawną odpowiedzią. Wystarczyły do tego dwa dodawania!

Można ciągnąć to dalej: $f(7)$ to 68, ponieważ $14+2$ to 16, a $16+52$ to 68; $f(8)$ to 86, ponieważ $16+2$ to 18, a $18+68$ to 86. Gdybyśmy potrzebowali tablicy wszystkich wartości f , wystarczy, że dodamy dwie liczby, aby uzyskać każdy kolejny wpis w tej tablicy. Żadnego mnożenia, żadnego odejmowania — tylko dwa proste dodawania!

Czy działa to dla sinusów? Przypuśćmy, że potrzebujemy tablicy sinusów w zakresie od 0 do $-\pi/2$ z krokiem 0,005. Przypuśćmy też, że używamy rozwinięcia Taylora stopnia 7. Wystarczy, że obliczymy osiem pierwszych wartości, a potem będziemy mogli po prostu dodawać różnice, aby uzyskać pozostałe wartości.

Obliczmy zatem wartość naszego wielomianu dla x równego 0,005:

$$0,005 - 0,005^3/6 + 0,005^5/120 - 0,005^7/5040$$

Zamiast rozwijać te ułamki w liczby dziesiętne i przez to tracić precyzję, zachowajmy je w nienaruszonej postaci do ostatniej chwili:

$$\begin{aligned} &0,005 - 0,000000125/6 + 3,125E-12/120 - 7,8125E-17/5040 \\ &5/1000 - 125/6000000000 + 3125/12000000000000000 - 78125/5040000000000000000 \\ &1/200 - 1/48000000 + 1/3840000000000 - 1/645120000000000000 \\ &322558656001679999/6451200000000000000 \end{aligned}$$

Dziesiętne rozwinięcie tego ostatniego ułamka to:

$$0,004999979166692708$$

Jest to wartość $\sin(0,005)$ z dokładnością do mniej więcej 12 cyfr.

Widzimy jednak teraz coś ważnego. Te mianowniki są duże, co sprawia, że ostateczny licznik ma mnóstwo cyfr znaczących (18). Oznacza to, że będziemy musieli wykonywać długie obliczenia i ciężko pracować, żeby nie stracić precyzji.

Praca z ułamkami o dużych mianownikach nie wydaje się dużo łatwiejsza niż obliczanie rozwinięcia Taylora. Spróbujmy więc czegoś innego. Pomnożmy wszystkie te ułamki przez 10^{30} i skróćmy mianowniki do rozsądniejszej postaci.

Teraz nasza pierwsza wartość to $314998687501640624023437500000/63$. Jeśli wykonamy to dzielenie i zignorujemy część ułamkową, otrzymamy następującą liczbę całkowitą: 4999979166692708317832341269. Jest ona bardzo bliska wartości $\sin(0,005)$ razy 10^{30} .

Zatem dzięki skróceniu mianowników możemy zignorować części ułamkowe bez znacznej utraty precyzji. Użyjmy teraz tej techniki, aby obliczyć siedem następnych sinusów i uzyskać w ten sposób pierwszych osiem:

```
314998687501640624023437500000/63
314990812550859250976562500000/63
314975062846169864257812500000/63
314951438781314260742187500000/63
314919940946892831054687500000/63
314880570130349793945312500000/63
314833327315953508789062500000/63
314778213684771866210937500000/63
```

Obliczmy teraz siedem pierwszych różnic:

```
-124999218751953125000000
-749985937589843750000000/3
-1124955469314453125000000/3
-1499896877210937500000000/3
-1874800787763671875000000/3
-2249657828394531250000000/3
-2624458627697265625000000/3
```

A następnie drugie, trzecie, czwarte, piąte i szóste różnice — które kończą się stałymi, tak jak powinny:

```
-374988281333984375000000/3
-124989843908203125000000
-124980469298828125000000
-374903910552734375000000/3
-124952346876953125000000
-124933599767578125000000
18749609375000000000/3
93746093750000000000
37497343750000000000/3
46869921875000000000/3
18747109375000000000

9374218750000000000/3
9373515625000000000/3
```

9372578125000000000/3
9371406250000000000/3

-234375000000000
-312500000000000
-390625000000000

-781250000000000
-781250000000000

Zatem siedem pierwszych różnic to:

314998687501640624023437500000/63
-124999218751953125000000
-374988281333984375000000/3
18749609375000000000/3
9374218750000000000/3
-234375000000000
-781250000000000

Z takimi dużymi licznikami i małymi mianownikami znów możemy bezpiecznie zignorować część ułamkową i rozważyć tylko część całkowitą, ponieważ powtarzające się cyfry na dalekich miejscach dziesiętnych nie narobią większych szkód, kiedy będziemy wykonywać dodawania. Zatem różnice te stają się następującymi liczbami całkowitymi:

4999979166692708317832341269
-124999218751953125000000
-124996093777994791666666
6249869791666666666
3124739583333333333
-234375000000000
-781250000000000

Teraz wystarczy dodawać te liczby, aby wypełniać każdy kolejny wiersz tabeli. Daje nam to bardzo sympatyczny zestaw sinusów pomnożonych przez 10^{30} :

0.005	4999979166692708317832341269
0.01	9999833334166664682539682538
0.01	14999437506328091099330357141
0.02	1999866669333079365079365078
0.025	24997395914712330651661706348
0.03	29995500202495660714285714282
0.035	34992854604336192599826388876
0.04	39989334186634158730158730124
0.045	44984814037660234235491071351
0.05	49979169270678323412698412546
0.055	54972275027067721183655753695
0.06	59964006479444571428571428114

Mniej więcej w taki sposób budowano tablice przed powstaniem automatycznych komputerów.

Mistrzowie matematyki odpowiedzialni za generowanie tablic obliczali wielomiany najlepiej pasujące do funkcji transcendentálnych, które chcieli przybliżyć. Wręczali te wielomiany

kilku wykwalifikowanym matematykom, którzy dzielili te funkcje na stosunkowo małe zakresy i obliczali tablice różnic dla każdego zakresu. Na koniec tablice różnic trafiały do armii kilkudziesięciu osób, których umiejętności były wystarczające do wykonywania prostych operacji dodawania i odejmowania. Osoby te, zwane „komputerami”⁶, wykonywały wszystkie dodawania, wypełniając pozycje tabeli w każdym zakresie.

Tysiące tysięcy wierszy. Dziesiątki tysięcy dodawań wykonywanych na długich rzędach cyfr. Płace były niskie, praca żmudna, a wszystko to robiono piórem na papierze⁷.

„Komputerów” zwykle dzielono na dwa zespoły, przydzielając każdemu z nich to samo zadanie. Jeśli oba idealnie wykonały swoją pracę, wyniki się zgadzały. I właśnie to sprawdzali Babbage i Herschel owego pamiętnego dnia latem 1821 roku.

Wizja Babbage’a

Frustracja Babbage’a i wynikające z niej marzenie o liczącej maszynie parowej są teraz zrozumiałe. Gdyby „komputerów” można było zastąpić niezawodną maszyną, on i Herschel już nigdy więcej nie musieliby wykonanych obliczeń sprawdzać ręcznie.

Co więcej, Babbage widział przyszłość, w której wykorzystanie mechanizmów obliczeniowych byłoby niezbędne dla rozwoju nauki.

W 1822 roku, wkrótce po swojej „parowej” skardze, proroczo napisał:

„Ośmielę się jeszcze przewidzieć, że nadejdzie czas, gdy kumulująca się praca, która bierze się z arytmetycznego zastosowania wzorów matematycznych, działając jako stale opóźniająca siła, ostatecznie utrudni postęp naukowy, chyba że zostanie opracowana ta⁸ lub jakaś równoważna metoda, która uwolni nas od przytłaczającego ciężaru liczbowych szczegółów”.

Niestety dla Babbage’a ten czas był odległy o ponad stulecie. Ale w końcu nadszedł. Jak zobaczysz w dalszych rozdziałach, uderzył wtedy ze zdwojoną siłą.

⁶ Na początku XIX wieku często były to francuskie fryzjerki, które straciły klientów, ponieważ ich klienci stracili głowy.

⁷ Pierwsza fabryka ołówków w Stanach Zjednoczonych zaczęła działać w 1861 r. Oczywiście byli poprzednicy, ale potrzeba było masowej produkcji, żeby ołówek stał się przedmiotem codziennego użytku.

⁸ Odnosi się to do pomysłów Babbage’a na maszynę różnicową.

Maszyna różnicowa

Maszyna, którą Babbage pierwotnie sobie wyobrażał, była dużym sumatorem napędzanym korbą. Miałaby sześć rejestrów po 20 cyfr dziesiętnych, które reprezentowałyby sześć różnic. Dzięki temu radziłaby sobie z wielomianami szóstego rzędu. Każdy cykl maszyny (każdy obrót korby) powodowałby dodawanie szóstego rejestru do piątego, następnie piątego do czwartego, potem czwartego do trzeciego i tak dalej. Dodawanie. I nic ponadto.

Maszyna składałaby się z 25 000 osobnych części, byłaby wysoka na 2,44 metra, długa na 2,13 metra i szeroka na 0,8 metra oraz ważyłaby około 4 ton.

Dlaczego dodawanie było tak trudnym zadaniem? Dlaczego jego realizacja wymagała maszyny o przemysłowej skali?

Problem był dwójaki. Przede wszystkim maszyna musiała być *dokładna*. Każda ruchoma część musiałaby być przytrzymywana w miejscu, chyba że jej ruch byłby akurat konieczny. Nie można było pozwolić na wibracje, tarcie ani żaden inny rodzaj pasożytniczego ruchu, który mógłby przypadkiem poruszyć część. Dlatego Babbage dodał wiele mechanizmów do blokowania i odblokowywania części, aby utrzymać integralność maszyny w tysiącach cykli obrotu korbą, które musiałaby wytrzymać.

Drugim problemem był szkolny problem przenoszenia. Kiedy 9 zmienia się w 0, przenosimy 1 do następnej pozycji po lewej stronie. W tradycyjnych mechanizmach z epoki robiło się to przez reprezentowanie każdej cyfry za pomocą koła, przy czym na każdym kole był zaczep, który poruszał następnym kołem, kiedy poprzednie przechodziło od 9 do 0. Zatem siła potrzebna do obrócenia koła zależała od liczby przeniesień. Dodawanie 1 do 999 999 999 999 wymagałoby *bardzo* dużej siły. Duże obciążenia powodują błędy i przyspieszają zużycie mechanizmów. Ponadto maszyna miała być ręcznie napędzana korbą, więc ramiona nieszczęsnego operatora doświadczałyby bardzo zróżnicowanych obciążeń.

Babbage wymyślił więc bardzo sprytny mechanizm zapamiętywania przeniesień i przekazywania ich jedno po drugim w innej części cyklu. Pamięć ta miała postać małej dźwigni związanej z każdym kołem. Dźwignia mogła znajdować się w jednym z dwóch stanów: przeniesienie lub brak przeniesienia. Przeniesienia były następnie przekazywane dalej przez zestaw prętów, z których każdy badał stan odpowiedniej dźwigni i dodawał 1 do koła powyżej, jeśli jego dźwignia była w stanie przeniesienia. Pręty te były ułożone w rodzaj spiralnej klatki schodowej, dzięki czemu każde przeniesienie miało miejsce po zakończeniu poprzedniego.

Wszystko to było bardzo pomysłowe.

Babbage był programistą. Program, który napisał, miał formę dźwigni, kół, przekładni zębatych i korb, ale bez wątpienia był to program. Maszyna Babbage'a wykonywała przepiękną procedurę kolejnych dodawań.

Kod, którego użylibyśmy dziś w tym programie, mógłby wyglądać mniej więcej tak:

```
(defn crank [xs]
  (let [dxs (concat (rest xs) [0])] (map + xs dxs)))
```

Tak, dwa wiersze kodu, około 74 znaków, zastępują 25 000 mechanicznych części o wadze 4 ton.

Ale może to nie jest sprawiedliwe. W końcu wykonuję ten kod na MacBooku Pro, a choć laptop ten waży niespełna kilogram, jest o wiele bardziej skomplikowany niż biedna maszyna różnicowa Babbage’a.

À propos, może zauważyłeś, że niektóre różnice w przykładzie z sinusami z poprzedniego podrozdziału były ujemne. A jednak maszyna Babbage’a mogła reprezentować tylko dodatnie liczby całkowite. Jak więc Babbage sobie z tym poradził?

Nie wiem na pewno, jak to zrobił, ale ja wykorzystałbym w tym celu uzupełnienie do 10. Jeśli masz tylko 20 cyfr i zignorujesz przeniesienie 20. cyfry, możesz reprezentować liczby ujemne przez ich „dziewiątkowanie”.

Na przykład 1 w uzupełnieniu do 10 to 99 999 999 999 999 999. Jeśli dodasz te dwie liczby i zignorujesz końcowe przeniesienie, otrzymasz 0. Zatem uzupełnienie 1 do 10 działa jak -1 .

Ogólna procedura jest taka, że bierzemy każdą cyfrę liczby i odejmujemy ją od 9, uzyskując w ten sposób odpowiednią cyfrę uzupełnienia. Kiedy wykonamy tę operację na wszystkich cyfrach, dodajemy 1 do końcowego wyniku.

Jeśli zatem mamy liczbę 31 415 926 535 887 932 384, możemy ją „dziewiątkować” i uzyskać uzupełnienie do 9, czyli 68 584 073 464 112 067 615, a następnie dodać 1, aby uzyskać uzupełnienie do 10, czyli 68 584 073 464 112 067 616. Kiedy dodamy te dwie liczby i zignorujemy końcowe przeniesienie, otrzymamy 0.

Tak więc uzupełnienie do 10 to efektywny sposób na utworzenie liczby ujemnej i zmianę odejmowania w dodawanie.

Notacja mechaniczna

Za tym, że Babbage był programistą, przemawia również to, iż napotkał problem, z którym wcześniej nie mierzył się żaden inżynier mechanik: dynamikę. Części jego maszyny poruszały się w skomplikowany sposób i w skomplikowanych momentach. Łączyły się też i rozłączały ze sobą w podobnie skomplikowany sposób. Skala złożoności mechanicznej była zupełnie nowa i wymagała jakiejś formy reprezentacji.

Babbage stworzył więc formalną notację dynamiki swojego urządzenia. Formalizm ten obejmował diagramy czasowe, logiczne diagramy przepływu i różnorodne konwencje symboliczne, które identyfikowały części w ruchu, w bezruchu i w każdym innym stanie potrzebnym do opisania intencji Babbage’a. Babbage uważał tę notację za „uniwersalny abstrakcyjny język” współoddziałujących części. Uważał, że można ją zastosować do dowolnej formy interakcji, mechanicznej lub innej.

Oto co powiedział na jej temat⁹:

„Trudność ogarnięcia umysłem wszystkich równoczesnych i następujących po sobie ruchów skomplikowanej maszyny i jeszcze większa trudność poprawnego synchronizowania zaplanowanych ruchów skłoniły mnie do poszukiwania metody, dzięki której mógłbym jednym rzutem oka wybrać dowolną część i w dowolnym momencie znaleźć jej stan ruchu lub spoczynku, jej związek z ruchami dowolnej innej części maszyny, a w razie potrzeby prześledzić źródła jej ruchu poprzez wszystkie jej kolejne stany aż do pierwotnej siły napędowej”.

Często mówi się, że Babbage nie uświadomił sobie, iż liczby, które obsługiwała jego maszyna, mogły reprezentować symbole innego formalizmu. Widzimy jednak, że Babbage nie miał problemu z wynalezieniem zupełnie nowego formalizmu dynamiki. Zatem operowanie symbolami bynajmniej nie było mu obce.

Imprezowe sztuczki

Kolejnym dowodem na to, że Babbage był programistą, była jego umiejętność programowania prototypowych mechanizmów w celu płatania salonowych figli kolacyjnym gościom.

Jak mogłaby wyglądać taka sztuczka? Powiedzmy, że Babbage zbiera grupę londyńskich celebrytów wokół małego prototypu maszyny w swoim salonie. Pokazuje im 0 w ostatnim rejestrze, a potem kręci korbą. Liczba wzrasta z 0 do 2. Pyta swoich gości, jaka liczba pojawi się po następnym zakręceniu korbą, a ci, którzy zgadują 4, są mile zaskoczeni. Pyta ponownie i wszyscy zgadują 6, potem 8, potem 10. Ale zanim się znudzą, kolejny obrót korby daje 42.

Gdy wiesz, jak działa maszyna różnicowa, nie jest trudno przygotować takie niespodzianki. W końcu sekwencja 0, 2, 4, 6, 8, 10, 42 ma zbiór różnic, które po dodaniu do siebie spowodują odtworzenie tej sekwencji.

Babbage następnie mówił gościom, że jego maszyna przestrzega ukrytych praw, które są znane tylko jemu. Twierdził, że „cud” liczby 42 przypomina takie cuda, jak rozstąpienie się wód Morza Czerwonego albo uzdrawianie chorych. Bóg jest bowiem programistą, który zbudował wszechświat zgodnie z ukrytymi prawami, które zna tylko On¹⁰.

⁹ Swade, s. 119.

¹⁰ Swade, s. 79.

Upadek maszyny

W 1823 roku Babbage, będąc szanowanym członkiem Astronomical Society i Royal Society oraz mając życzliwe kontakty w Parlamencie, doprowadził do powołania finansowanej przez rząd komisji do spraw budowy swojej maszyny.

Decyzja o finansowaniu była kontrowersyjna. Niektórzy uważali, że jego urządzenie będzie przydatne, inni, że koszt nie jest wart korzyści. W końcu bezrobotne fryzjerki nie były drogie — i potrafiły dodawać.

Ale Babbage był zagorzałym orędownikiem wynalazku i charyzmatycznym mówcą. Często opowiadał ludziom o zaletach maszyny i niewiarygodnych mocach, jakie miała posiadać. Twierdził nawet, że sam nie znał wszystkich tych mocy. Potrafił utrzymać publiczność w napięciu swoim entuzjazmem dla projektu.

I rzeczywiście jest coś magicznego w pomysle, że przez przyłożenie zwykłej fizycznej siły maszyna mogłaby zrobić coś, co wcześniej należało do domeny *myślenia*. W tamtych czasach maszyna, która potrafiłaby myśleć, zadziwiłaby każdego jeszcze bardziej niż dzisiaj.

Dzięki tym wysiłkom i entuzjastycznemu wsparciu przyjaciół i zwolenników Babbage zatriumfował i zapewnił sobie obietnicę finansowania. Finansowanie to zaczęło się od kwoty 1500 funtów i z czasem wzrosło do ponad 17 000 funtów. Ponadto Babbage włożył w to przedsięwzięcie sporo własnych pieniędzy.

Prace ruszyły pełną parą i zaczęły powstawać prototypy. W ciągu następnej dekady wykonano wiele części i wyprodukowano małe maszyny demonstracyjne. Jednak skala urządzenia była duża, a Babbage łatwo się rozpraszał i raczej nieprzyjemnie się z nim współpracowało. Był drażliwy i dumny, nigdy nie zapominał zniewagi lub urazy i narobił sobie wrogów. Po dekadzie opóźnień, sporów z wykonawcą, przerw w pracy i znacznych przekroczeń kosztów finansowanie ostatecznie wycofano.

Przyjaciele i zwolennicy Babbage'a, którzy początkowo go wsparli, byli rozczarowani i zawstydzeni tą ostateczną porażką. Wydano sporo publicznych pieniędzy bez żadnego namacalnego efektu. Było mało prawdopodobne, żeby którykolwiek z nich poparł go ponownie.

Tak więc maszyna Babbage'a nigdy nie została zbudowana¹¹. Oczywiście skonstruowano sporo części. W rzeczywistości maszyna, którą Babbage zabawiał swoich gości, została zbudowana z części przeznaczonych do maszyny pełnowymiarowej.

¹¹ Znacznie ulepszony projekt, maszyna różnicowa 2, został ostatecznie zbudowany w London Science Museum pod koniec lat 80. i na początku lat 90. Jest tam wystawiana i działa tak, jak zaprojektował ją Babbage, choć debugowanie jej było niemałym wyzwaniem.

Ostatecznie jednak, za radą George’a Bidella Airy’ego, Królewskiego Astronoma, rząd odmówił dalszego finansowania. List z biura premiera brzmiał następująco:¹²

„Projekty pana Babbage’a wydają się tak kosztowne, ostateczny sukces tak wątpliwy, a wydatki tak duże i zupełnie niemożliwe do obliczenia, że rząd nie znajduje uzasadnienia dla wzięcia na siebie jakiegokolwiek dalszej odpowiedzialności”.

Zatem po dekadzie prac i wydaniu 17 000 funtów projekt zarzucono.

Moim skromnym zadaniem — choć myślę, że Babbage prawdopodobnie by zaprotestował — ostatecznie projekt zakończył się niepowodzeniem w dużej mierze dlatego, że uwagę Babbage’a nieustannie przyciągały inne, bardziej ambitne pomysły. Doprowadzanie projektów do końca nie było dla niego tak pasjonujące, jak ich rozpoczynanie.

Argument technologiczny

Niektórzy twierdzą, że niepowodzenie w budowie maszyny różnicowej wynikało z tego, iż technologia obróbki metali z początku XIX wieku nie była wystarczająco zaawansowana do tworzenia części z wymaganą precyzją. Nie ma jednak żadnych dowodów na poparcie tego twierdzenia. Ocalałe części są bardzo dokładne, istnieją też prototypy maszyn, które działają do dziś.

Gdyby Babbage miał wolę, skupienie i zasoby, aby ukończyć swoją maszynę, najprawdopodobniej działałaby ona zgodnie z projektem. Chociaż, jak się wkrótce przekonamy, sprawienie, aby maszyna faktycznie działała, to zupełnie co innego niż jej zbudowanie.

Maszyna analityczna

A mimo to mogę przeprowadzić w moich obwodach nieskończone strumienie delta prawdopodobieństwa przyszłości, aby ujrzeć, że pewnego dnia musi pojawić się komputer, któremu nawet parametrów operacyjnych nie jestem godzien obliczyć, lecz moim ostatecznym przeznaczeniem jest zaprojektować go. (tłum. A. Banaszak)

— Głęboka Myśl, Autostopem przez galaktykę

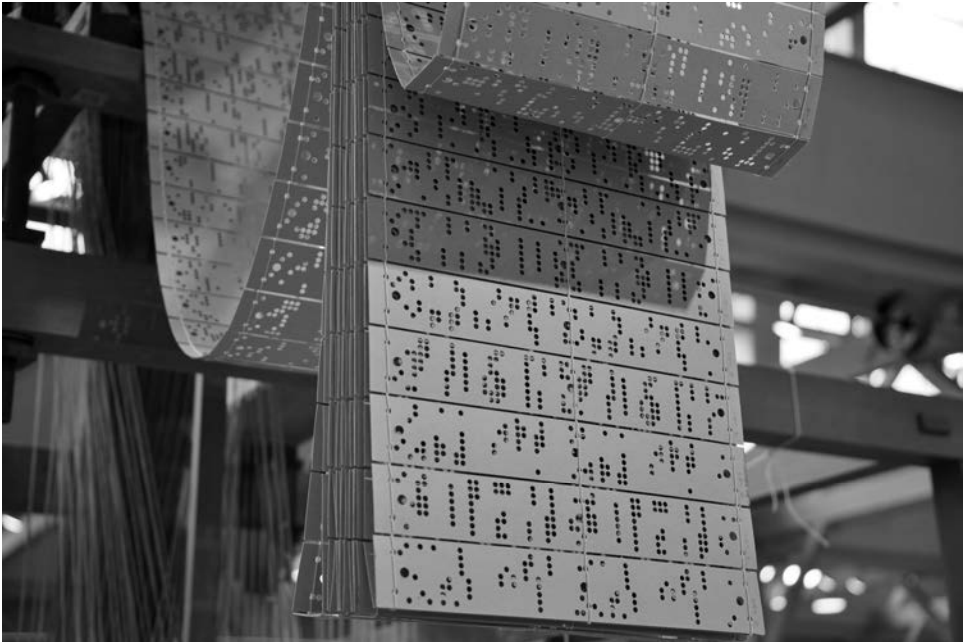
Co takiego przeszkodziło Babbage’owi w ukończeniu maszyny różnicowej?

Wyobraź sobie maszynę rozmiarów lokomotywy. Zamiast 6 kolumn po 20 cyfr ma ona 1000 kolumn po 50 cyfr. Ma mechaniczną szynę, która przenosi wartości tych kolumn do dwóch kanałów wejściowych młyna. Młyn może dodawać, odejmować, mnożyć i dzielić

¹² Swade, s. 176.

te liczby z pojedynczą lub podwójną (100-cyfrową) precyzją. Szyna następnie przenosi wynik z powrotem do liczącego 1000 kolumn magazynu.

Wyobraź sobie, że maszyną tą sterują instrukcje wybite na kartach dziurkowanych, które są połączone w łańcuch, jak karty drukowane na krośnie żakardowym (rysunek 2.5). Instrukcje te nakazują szynie pobrać wartości z określonych kolumn w magazynie i przenieść je szyną na wejścia młyna. Nakazują młynowi operować na wartościach dostarczonych przez szynę. Nakazują szynie przenieść wyniki z młyna z powrotem do magazynu.



Rysunek 2.5. Krosno żakardowe

Wczytują stałe do pamięci. Nakazują wydrukować wartości na drukarce albo wykresić krzywe na ploterze, albo po prostu uderzyć w dzwonek.

Co najważniejsze, są karty, które nakazują czytnikowi kart przejść o n kart do przodu lub do tyłu, *jeśli* ostatnia operacja spowodowała przepełnienie.

Z dwoma ogromnymi wyjątkami jest to architektura współczesnego komputera. Jakie to wyjątki? Po pierwsze, program jest przechowywany na kartach, a nie w liczącym 1000 kolumn magazynie, a po drugie, maszyna jest całkowicie mechaniczna, napędzana... tak, parą.

Babbage dołożył wszelkich starań, aby młyn działał wydajniej od maszyny różnicowej. Ostatecznie oszacował, że 50-cyfrowe liczby będzie można dodawać lub odejmować w czasie krótszym niż sekunda. Babbage zaprojektował młyn tak, aby mnożył za pomocą techniki

przesuwania i dodawania, a dzielił za pomocą techniki przesuwania i odejmowania. Dzięki temu nawet mnożenia i dzielenia o podwójnej precyzji zajmowałyby tylko minutę¹³.

Wyobraź sobie to steampunkowe monstrum w akcji. Wyobraź sobie zgrzytanie, brzęczenie i warczenie jego organów wewnętrznych sterowanych skrobaniem kart przez czytniki. Wyobraź sobie śledzenie wartości przepływających z magazynu przez szynę do młyna i z powrotem. Na pewno widzisz *ten ruch*!

Wyobraź sobie młyn, który przesuw a dodaje, przesuw a dodaje, wystukuje iloczyny i wyrzękuje ilorazy i reszty. Wyobraź sobie, że obserwujesz program, który działa, działa, działa, podczas gdy maszyna przesuw a to tu, to tam wartości w magazynie i młynie, od czasu do czasu drukując liczbę, przesuwając rysik na ploterze lub dzwoniąc dzwonkiem.

Czy zdołałbyś oderwać od niej wzrok?

Babbage nie potrafił. Zobaczył działanie maszyny w swoim umyśle i zrozumiał implikacje. Powiedział, że „teraz cała arytmetyka wydaje się być w zasięgu mechanizmu”¹⁴. Uważał, że maszynę można by nawet zaprogramować do gry w szachy¹⁵. W 1832 roku napisał w *Życiu filozofa*: „W każdą grę wymagającą umiejętności może grać automat”.

Oczywiście maszyny tej nigdy nie zbudowano. Babbage skonstruował małe prototypy młyna i majstrował przy różnych mechanizmach, które chciał wypróbować. Jednak w głębi duszy wiedział, że nigdy nie znajdzie niemal nieskończonych funduszy, czasu i energii niezbędnych do zbudowania tej bestii.

Ostatecznie całe to majsterkowanie i optymalizowanie skłoniły go do zastanowienia się nad niedociągnięciami maszyny różnicowej i zaprojektowania maszyny różnicowej 2, która miała jedną trzecią części, była trzy razy szybsza i miała ponad dwukrotnie większą pojemność niż pierwowzór. Nawet wtedy duch Moore’a¹⁶ miał się dobrze.

Naturalnie Babbage nigdy nie zbudował również tej maszyny; ale to właśnie to urządzenie, maszynę różnicową 2, zbudowano w londyńskim Science Museum pod koniec ubiegłego wieku¹⁷.

¹³ Swade, s. 111.

¹⁴ Swade, s. 91.

¹⁵ Swade, s. 179.

¹⁶ Gordon Moore, twórca prawa Moore’a, które przewiduje, że gęstość obwodów scalonych będzie podwajać się co dwa lata.

¹⁷ Swade, p. 221.

Symbole

Musimy tu jeszcze raz odrzucić twierdzenie, że Babbage nie uświadamiał sobie, iż jego mechanizmy mogłyby operować symbolami. Skoro uważał, że maszynę analityczną można by zaprogramować do gry w szachy, to musiał wiedzieć, jak symbolicznie przedstawić szachownicę, figury i ruchy.

Babbage rozumiał też, że poprawnie zaprogramowana maszyna mogłaby przetwarzać algebrę symboliczną. Ujął to w następujący sposób¹⁸:

„Tego dnia po raz pierwszy miałem ogólny, ale bardzo niejasny pomysł zbudowania maszyny, która mogłaby wykonywać przekształcenia algebraiczne — bez żadnego odniesienia do wartości liter”.

Babbage był programistą. Rozumiał związek między symbolami a liczbami i to, że każda maszyna, która może operować liczbami, mogłaby, dzięki temu związkowi, operować również symbolami.

Ada, hrabina Lovelace

Wpadłszy w złe dni na złe języki, zwrócił się Milton do mściciela, Czasu...

Być może zły to był dzień, w którym urodził się George Gordon Byron. Był utalentowanym i płodnym pisarzem i poetą, jednym z najlepszych w Anglii. Napisał takie arcydzieła jak *Don Juan* i *Melodie hebrajskie*. W wieku 10 lat odziedziczył baronię Rochdale i w ten sposób został lordem Byronem.

Byron nie był jednak wzorem cnót. Był porywczym kobieciarzem, który spłodził wiele nieślubnych dzieci, w tym z przyrodnią siostrą Mary Shelley¹⁹ i z własną przyrodnią siostrą Augustą Marią Leigh.

Aby złagodzić presję narastających długów, szukał odpowiedniej partii. Wśród licznych celów tych poszukiwań była Annabella Milbanke²⁰, prawdopodobna dziedziczka bogatego wuja. Chociaż początkowo mu odmawiała, ostatecznie uległa i poślubiła lorda Byrona w styczniu 1815 roku. Miał z nią swoje jedyne prawowite dziecko. Dziewczynka urodziła się w grudniu i otrzymała imiona Augusta Ada.

¹⁸ Swade, s. 169.

¹⁹ Mary odwiedzała dom Byrona w pobliżu Jeziora Genewskiego w Szwajcarii latem 1816 r. — „roku bez lata”. Kula ziemską pogrążyła się wówczas w niszczycielskiej fali zimna w wyniku eksplozji wulkanu Mount Tambora. Podczas deszczowych dni i nocy Mary i Byron siadywali przy ognisku i czytali opowieści o duchach wraz z elitarną grupą przyjaciół. Byron rzucił im wyzwanie, żeby napisali opowieść o duchach, co zainspirowało Mary do napisania *Frankensteina*.

²⁰ Anne Isabella Milbanke, ale nazywaną ją Annabellą.

Byron nie był zadowolony z jej narodzin, spodziewał się bowiem „wspaniałego chłopca”. Czy była to zniewaga, czy zaszczyt, nazwał ją na cześć swojej kochanki i przyrodniej siostry, ale zawsze zwracał się do niej imieniem Ada.

Biografka²¹ Byrona opisała kiedyś jego małżeństwo z Annabellą jako jedno z najbardziej nieszczęśliwych w historii²². Zachowanie Byrona było okropne. Kontynuował romans ze swoją przyrodnią siostrą i nie stronił od seksualnych eskapad z innymi kobietami, w tym z kilkoma znanymi aktorkami.

Byron napastował Annabellę przy czterech różnych okazjach, więc kazała swoim służącym zamykać przed nim drzwi.

Kontynuował swoje pijackie ekscesy, a w końcu próbował nawet eksmitować Annabellę z ich domu. Uznała go za szalonego i odeszła, zabierając ze sobą pięcioletnią Adę.

Był to skandal szeroko dyskutowany w londyńskim towarzystwie. W kwietniu Byron uciekł na kontynent, by już nigdy nie wrócić i nigdy więcej nie zobaczyć swojej córki Ady.

Ale reakcja społeczeństwa na Adę była przeciwieństwem zaniedbania ze strony jej ojca. Ludzie bardzo się nią interesowali ze względu na słynnego ojca uwodziciela.

Annabella była wykształcona matematycznie i wykorzystała to wykształcenie, aby odwrócić uwagę Ady od ojca i jego szaleństwa. Ada miała dar i pokochała matematykę, być może bardziej niż jej matka, ale nigdy nie przestała interesować się ojcem. Nadała swoim dzieciom jego imiona i poprosiła, żeby pochować ją w pobliżu jego grobu.

Jej pamięć o ojcu mogły wzmacniać częste nieobecności Annabelli i jej pozorny brak uczuć. Kiedy Annabella mówiła o Adzie, czasami używała zaimka „to”. Przez większość czasu pozostawiała Adę pod opieką babci, swojej matki²³, która ją wychowywała i rozpieszczała.

Ada była chorowitym dzieckiem. Cierpiała na bóle głowy i niewyraźne widzenie. W okresie dojrzewania doznała paraliżu po zachorowaniu na odrę i przez większą część roku była przykuta do łóżka. Kontynuowała jednak naukę matematyki — być może aż nadto gorliwie. Gdy miała 17 lat, próbowała uciec z domu z jednym ze swoich nauczycieli²⁴.

Inna nauczycielka Ady, Mary Somerville, zapoznała ją z wieloma ważnymi figurami ówczesnego świata matematyki i nauki — w tym z Charlesem Babbage’em.

²¹ Benita Eisler, *Byron: Child of Passion, Fool of Fame*.

²² Swade, s. 156.

²³ Judith Lamb Noel.

²⁴ Z Williamem Turnerem. Ada twierdziła, że związek nie został skonsumowany.

Później, na jednym z licznych przyjęć u Babbage’a, Ada zobaczyła prototyp maszyny różnicowej i zafascynowała się jej działaniem. Od tego czasu często odwiedzała Babbage’a, aby rozmawiać o jego maszynach i innych planach. Babbage opowiedział jej o swoim ambitnym celu, którym była maszyna analityczna. Ada rozkoszowała się jej zawłościami i możliwościami.

Połknęła bakcyła. Została programistką.

W wieku 19 lat poślubiła Williama Kinga, ósmego lorda Ockhama i pierwszego hrabiego Lovelace. I tak Ada została hrabiną Lovelace. Mimo słabego zdrowia oraz obowiązków związanych z dziećmi i domem nadal zgłębiała zarówno matematykę, jak i idee Babbage’a. Poprosiła Babbage’a, żeby polecił jej korepetytora, a on powierzył tę rolę Augustusowi De Morganowi²⁵.

Niedługo potem Babbage otrzymał zaproszenie od matematyka Giovanniego Plany, aby przedstawić koncepcję swojej maszyny analitycznej na zjeździe włoskich naukowców w Turynie. Babbage z radością się zgodził. Miał to być pierwszy i ostatni raz, kiedy publicznie opisał ten wspaniały pomysł.

Wykład został dobrze przyjęty, a Plana obiecał opublikować raport z sesji. Babbage czekał na ten raport prawie dwa lata.

Opóźnienie było prawdopodobnie wynikiem komplikacji w życiu Plany, a może również tego, że Plana nie był tak entuzjastycznie nastawiony, jak się wydawało. Ostatecznie więc powierzył to zadanie Luigiemu Menabrei, 31-letniemu inżynierowi, który uczestniczył w sesji. Raport, napisany po francusku, został opublikowany w szwajcarskim czasopiśmie w 1842 roku.

Charles Wheatstone²⁶, przyjaciel Ady i Babbage’a, przeczytał raport i zasugerował, że wspólnie z Adą mogłby opracować angielskie tłumaczenie, by opublikować je w „Scientific Memoirs”²⁷. Ada zgodziła się i wykorzystała swoją biegłą znajomość francuskiego i głęboką wiedzę o maszynie analitycznej. Pod czujnym okiem Wheatstone’a tłumaczenie zostało ukończone i zaprezentowane Babbage’owi jako niespodzianka — wysiłek przyjaciół na rzecz jego idei.

Babbage był zadowolony, ale powiedział Adzie, że z pewnością potrafiłaby napisać oryginalną pracę na ten temat, a następnie zasugerował, aby wykorzystała swoją wiedzę i dodała notatki do tłumaczenia.

²⁵ Tak, *temu* De Morganowi. Od $AB = \sim(\sim A + \sim B)$.

²⁶ Tak, *ten* Wheatstone. Od mostka Wheatstone’a.

²⁷ Czasopismo publikujące zagraniczne artykuły naukowe.

Zachwyceni tą perspektywą, Ada i Babbage rozpoczęli intensywną współpracę obejmującą wizyty, listy i wiadomości. Ada „pracowała z szaloną energią, [stając się] wymagającą, apodyktyczną, kokieterijną i drażliwą”²⁸. Im więcej pracowała, tym bardziej rósł jej entuzjazm.

I tu sprawy przybierają dziwny obrót. Ada, hrabina Lovelace, miała skłonność do manii. Kiedyś w liście do matki zawarła dziwacznie monomaniakalny opis samej siebie. Zawierał on następujące fragmenty²⁹:

„Uważam, że posiadam zupełnie wyjątkową kombinację cech, które *predestynują* mnie do tego, żebym została wybitną odkrywczynią *ukrytych realiów natury*”.

„[...] dzięki pewnej osobliwości mojego układu nerwowego *domyślam się* pewnych rzeczy, których nie dostrzega nikt inny [...]”.

„[...] moje ogromne zdolności rozumowania [...]”.

„[Mam] moc nie tylko wkładania całej swojej energii i jestestwa w cokolwiek wybiorę, ale także kierowania ich na dowolny temat lub ideę, ogromny aparat złożony ze wszystkich pozornie nieistotnych i zewnętrznych źródeł. Mogę rzucać *promienie* z każdej strony wszechświata w jedno ogromne ognisko [...]”.

O swoich listach ostatecznie powiedziała, że choć wydają się *szalone*, to są „bodaj najbardziej logicznymi, trzeźwymi i chłodnymi utworami, jakie kiedykolwiek napisałam; wynikiem wielu dokładnych, rzeczowych przemyśleń i studiów”.

W świetle tego rozważmy poniższe fragmenty listów, które pisała do Babbage’a w ferworze prac nad notatkami — listów, które czasami podpisywała „Twoja Pani Wróżka”:

„Im więcej studiuję, tym bardziej mój nienasycony geniusz rwie się do nauki”³⁰.

„*Nie* wierzę, że mój ojciec był (lub mógłby być) takim *poetą*, jaką ja będę *analityczką* (i metafizyczką)”.

Koniec końców powstało siedem notatek oznaczonych literami od A do G. Łącznie mają one trzykrotnie większą objętość niż sam artykuł. Zostały opublikowane w 1843 roku. Są błyskotliwe. I wylewne. Na przykład w notatce A Ada pisze:

„Maszyna analityczna nie ma wiele wspólnego ze zwykłymi »maszynami liczącymi«. Zajmuje całkowicie odrębną pozycję, a kwestie, które podpowiada, są najciekawsze w swojej naturze. Zapewnia bowiem mechanizm łączenia ze sobą ogólnych symboli”.

Z powodu tego zdania, a także kilku innych wzmianek o zdolności maszyny do reprezentowania symboli, a nie tylko wartości czysto liczbowych, Ada jest często nazywana pierwszym programistą.

²⁸ Swade, s. 161.

²⁹ Swade, s. 158.

³⁰ Swade, s. 161.

Pierwszy programista?

Jeśli przeczytasz notatki, nie będziesz mieć wątpliwości, że Ada Lovelace była programistką. Rozumiała maszynę. Była nią wręcz oczarowana. Potrafiła wyobrazić sobie jej działanie i przesłedzić wykonanie programu. Gdyby mogła jej dotknąć, prawdopodobnie opanowałaby jej obsługę.

Wyobraź sobie, że wiesz — dobrze wiesz! — co potrafi ta maszyna, a jednocześnie wiesz, że *nigdy* nie zobaczysz, jak działa — *w ogóle* nigdy jej nie zobaczysz. Jakaż to musiała być frustrująca mieszanka radości i rozczarowania, wielkiej wizji i rozpaczliwej nadziei!

Ale pomimo swojej kompetencji Ada, hrabina Lovelace, *nie* była pierwszym programistą. Nie ulega wątpliwości, że Babbage był przed nią i miał równą zdolność dostrzeżenia symbolicznej natury maszyny. Nie ma żadnego znaczącego spostrzeżenia, którego dokonałaby ona, a on nie.

Owszem, notatki są błyskotliwe. Owszem, Ada formalnie opisała programy, które mogłaby wykonywać maszyna. I choć znalazła usterkę w jednym z nich, to nie napisała tych programów — zrobił to Babbage.

Co więcej, nie ulega wątpliwości, że Ada nie jest jedynym autorem notatek. Współpraca z Babbage'em była tak intensywna, że jest to po prostu niemożliwe.

Jednak ze względu na tę współpracę *możemy* powiedzieć coś innego: Ada, hrabina Lovelace, może nie była pierwszym programistą, ale Ada i Babbage niemal na pewno byli *pierwszymi programistami pracującymi w parze*.

Tylko dobrzy umierają młodo

Dziewięć lat później, po długich cierpieniach fizycznych i emocjonalnych, Ada uległa rakowi szyjki macicy w wieku 36 lat. Została pochowana zgodnie z życzeniem: u boku ojca, który ją porzucił.

Mieszane zakończenie

Ostatecznie wysiłki Babbage'a i Lovelace przyniosły mieszane rezultaty. W swoim czasie spełzyły na niczym. Ada nic więcej nie opublikowała, a Babbage nie podjął dalszych prób przedstawienia swoich koncepcji. Wielka wizja pozostawała w zapomnieniu przez stulecie.

Chciałoby się rzec, że koncepcje tej pary programistów były iskrą, która rozpałała erę informacji — że pionierzy, którzy pojawili się ponad wiek później, inspirowali się ich pismami i wizjami. Ale niestety nie jest to do końca prawda. Jeśli ci późniejsi pionierzy

w ogóle ich przywołują, to jako wspomnienie lub ukłon w stronę podobnie myślących ludzi z za przepaści czasu.

Babbage ze swej strony odpowiedział na ten ukłon wiadomością wysłaną w przyszłość w 1851 roku. Wiadomością, w której pobrzmiewa nuta bólu i rozczarowania, jakie musiał odczuwać, wiedząc, że jego wielkie idee zostały zignorowane przez współczesnych:

„Pewność, że przyszła epoka naprawi niesprawiedliwość teraźniejszości, i świadomość, że im bardziej odległy jest dzień przygotowań, tym bardziej wyprzedza swoich współczesnych, podtrzymują go na duchu w obliczu szyderstw ignorantów lub zazdrości rywali”.

Jak przekonamy się w kolejnych rozdziałach, późniejsi pionierzy wypowiadali się z wielkim szacunkiem o Babbage’u i Lovelace. Zobaczymy nawet swego rodzaju powtórzenie ich relacji we współpracy Howarda Aikena i Grace Hopper, którzy zbudowali i zaprogramowali elektromechaniczny odpowiednik maszyny analitycznej Babbage’a: Harvard Mark I. Mimo to byłoby zbyt naciągane stwierdzenie, że Babbage i Lovelace w jakikolwiek znaczący sposób ukierunkowali ich prace lub na nie wpłynęli.

Babbage nie doprowadzał rzeczy do końca. Rozpoczął budowę maszyny różnicowej. Rozpoczął budowę maszyny analitycznej. Rozpoczął budowę maszyny różnicowej 2. Rysował projekty. Majstrował przy kawałkach i elementach. Nawet składał fragmenty maszyn, ale nigdy nie skończył żadnej z nich. Jego współcześni narzekali, że chętnie pokazywał im jeden pomysł, lecz potem następny, lepszy, i następny, jeszcze lepszy, używając każdego jako wymówki, która uzasadniała, dlaczego poprzedni pozostaje niedokończony.

Gdyby Babbage dokończył pierwszy projekt maszyny różnicowej, kto wie, co by się stało. Teraz wiemy, że maszyna by zadziałała. Czy ten sukces mógłby doprowadzić do powstania innych, wspanialszych maszyn? Czy Babbage w końcu zobaczyłby swoją wymarzoną maszynę analityczną w takiej czy innej formie?

Realizacja maszyny różnicowej 2

Gdyby Babbage ukończył tę pierwszą maszynę, z pewnością odkryłby, że nie pomyślał o żadnych środkach debugowania i testowania podczas montażu.

Pod koniec lat 80. i na początku lat 90. XX wieku w londyńskim Science Museum powstał działający egzemplarz maszyny różnicowej 2. To piękna maszyna, *lśniąca metalowa rama* z mosiądzu i stali. Gdy kręci się korbą, świdrokrętnie wężają kolumny liczbowe, a smukwiny wrzeczona dodają przeniesienia do sum. Oglądanie tego to prawdziwa rozkosz.

Ale opowieści osób, które ją zbudowały, są pełne frustracji.

Po złożeniu maszyna po prostu się zaciniała, kiedy obrócono korbą o więcej niż stopień lub dwa. Każda część musiała być idealnie wyrównana z każdą inną częścią zgodnie z zależnościami

czasowymi maszyny, czego Babbage nie uwzględnił w swoim projekcie. Nie jest jasne, czy w ogóle przewidział ten problem.

Stopniowe debugowanie, wyrównywanie i naprawianie maszyny zajęło 11 miesięcy. Czasem wymagało to małego przekręcenia korbą, aż się zacięła, a następnie grzebania w trzewiach maszyny śrubokrętami lub szczypcami w celu znalezienia odrobiny luzu w jednym miejscu albo braku luzu w drugim. Czasem trzeba było umyślnie zepsuć jakąś część, żeby odnaleźć źródło oporu. Czasem konieczne były nawet pomniejsze zmiany projektu.

Oto co o projekcie Babbage'a miał do powiedzenia Doron Swade, kustosz Science Museum, który kierował projektem od początku do końca:

„Babbage nie pomyślał o debugowaniu. Nie ma łatwego sposobu na odizolowanie jednej sekcji maszyny od drugiej w celu zlokalizowania miejsca zacięcia. Cała maszyna jest jedną monolityczną, skręconą „na sztywno” jednostką. Drażki napędowe i złącza są trwale przytwierdzone lub przynitowane i trudno je zdemontować po złożeniu”.

Jednak ostatecznie, po wyrównaniu wszystkich 4000 części i zainstalowaniu drobnych ulepszeń, maszyna zaczęła działać idealnie.

Istnieją naprawdę hipnotyzujące filmy, które pokazują maszynę w działaniu. Na początek polecam przeszukanie YouTube'a. (Kiedy pisałem tę książkę, na YouTube, na kanale Computer History Museum, dostępny był film *The Babbage Difference Engine #2 at CHM*, opublikowany 17 lutego 2016 r.).

Podsumowanie

Babbage był wynalazcą, majsterkowiczem, wizjonerem i... programistą. Niestety, jak wielu z nas, pozwolił, żeby lepsze stało się wrogiem dobrego. Jak wielu z nas, był zbyt pewny swoich projektów i poświęcał mało uwagi przyrostowym ulepszeniom. Jak wielu z nas, łatwo dawał się porwać idei i chętnie realizował ją w 80 procentach, ale tracił entuzjazm, kiedy przychodziło do ostatnich 20 procent, które wymagają 80 procent wysiłku.

Edison podobno mawiał, że wynalazek to 1 procent inspiracji i 99 procent potu. Babbage był świetny w 1 procencie, ale nigdy nie poradził sobie z tymi 99. Lubił rozmyślać. Lubił nawet budować fragmenty. Lubił rozmawiać o tym, co wymyślił, i demonstrować fragmenty, które zbudował. Ale gdy trzeba było naprawdę ciężko popracować, żeby coś dokończyć, bardziej interesowała go *Następna Wielka Rzecz*[™].

Źródła

Douglas Agams, *The Hitchhiker's Guide to the Galaxy*, Pan Books, 1979.

Kurt W. Beyer, *Grace Hopper and the Invention of the Information Age*, MIT Press, 2009.

Benita Eisler, *Byron: Child of Passion, Fool of Fame*, Knopf, 1999.

Amy Jollymore, *Ada Lovelace, an Indirect and Reciprocal Influence*. Artykuł opublikowany przez O'Reilly Media, Inc., 14 października 2013 r., www.oreilly.com/content/ada-lovelace-an-indirect-and-reciprocal-influence.

Maboth Moseley, *Irascible Genius: The Life of Charles Babbage*, Hutchinson, 1964.

Robert Scoble, *A Demo of Charles Babbage's Difference Engine*, 24:09. Opublikowano na YouTube 17 czerwca 2010 r. (film był dostępny w czasie pisania tej książki).

Doron Swade, *The Difference Engine*, Penguin Books, 2000.

University of St Andrews, *The early history of computing | Professor Ursula Martin (Lecture 1)*, 1:01:46. Opublikowano na YouTube 26 lutego 2020 r. (film był dostępny w czasie pisania tej książki).

Wikipedia: *Sketch of the Analytical Engine Invented by Charles Babbage*, L. F. Menabrea, translated and annotated by Ada Augusta, the Countess of Lovelace https://en.wikisource.org/wiki/Scientific_Memoirs/3/Sketch_of_the_Analytical_Engine_invented_by_Charles_Babbage,_Esq.

Inne ważne materiały to m.in. strony Wikipedii poświęcone Babbage'owi, maszynie różnicowej, maszynie analitycznej, Adzie Lovelace i lordowi Byronowi.

PROGRAM PARTNERSKI

— GRUPY HELION —



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion

My, programiści

Jeśli oczekuje się, że maszyna będzie nieomylna, nie może być również inteligentna.

Alan Turing, wykład z 1947 roku

Poznaj fascynującą drogę, jaką przeszła informatyka — od pierwszego kodu do dziś, epoki sztucznej inteligencji. Ta książka to nie tylko chronologiczny zapis historii, ale też opowieść o wizji, marzeniach i potencjale rozwoju. Charles Babbage zaprojektował maszynę analityczną, ale to Ada Lovelace jako pierwsza zrozumiała, że może w niej tkwić coś więcej: dostrzegła potencjał twórczego myślenia. Zapisała pierwsze linijki kodu, zanim w ogóle powstały komputery. Tak rozpoczęła się nowa era — era ludzi, których geniusz i wytrwałość miały moc zmieniania świata.

Poznaj historię programowania i przekonaj się, jak fascynujący jest świat koderów, od Charlesa Babbage'a i Ady Lovelace po Alana Turinga, Grace Hopper i Dennisa Ritchiego; od przełomowych bitów i bajtów po przejmujące ludzkie historie. Przekonaj się, że droga do sukcesu często bywa wyboista, triumf poprzedzają porażki, a depresja i kpiny ze strony innych to codzienność wielu pionierów. Strona po stronie odkrywaj nieoczywiste prawdy o technologii i o ludziach, których kod tworzy współczesny świat.

W książce:

- ▶ początki programowania
- ▶ pionierzy kodowania, ich motywacje i trudności, jakie musieli pokonać
- ▶ największe przełomy: od wczesnych assemblerów po języki obiektowe
- ▶ druga wojna światowa a rozwój informatyki
- ▶ kwestia etyki w rozwoju sztucznej inteligencji

Robert C. Martin (dla przyjaciół: Wujek Bob) programuje od 1970 roku. Jest założycielem firmy Uncle Bob Consulting LLC, która pracuje dla największych firm z całego świata. Opublikował dziesiątki artykułów i napisał kilka kultowych książek. Jest też twórcą serii filmów edukacyjnych dostępnych na stronie *cleancoders.com*. Był pierwszym przewodniczącym organizacji Agile Alliance.

Helion 

 helion.pl

 **HELION S.A.**
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI
Sięgnij po więcej! ▶



ISBN 978-83-289-2650-9



9 788328 926509

Cena: 89,00 zł



Pearson
Addison-Wesley