

Auf einen Blick

| | |
|---|------------|
| Einleitung | 19 |
| Teil I: Techniken und Technologien | 25 |
| Kapitel 1: Wie das Coding von KI profitiert. | 27 |
| Kapitel 2: Maschinelles Lernen und Deep Learning | 53 |
| Kapitel 3: KI-Coding-Tools. | 73 |
| Kapitel 4: Coding mit Chatbots | 97 |
| Teil II: KI fürs Coding einsetzen | 127 |
| Kapitel 5: Der Weg vom Plan zum Prototyp. | 129 |
| Kapitel 6: Code formatieren und verbessern | 157 |
| Kapitel 7: Fehler finden und beseitigen | 183 |
| Kapitel 8: Code übersetzen und optimieren | 211 |
| Teil III: Code testen, dokumentieren und pflegen | 227 |
| Kapitel 9: Code testen | 229 |
| Kapitel 10: Code dokumentieren | 253 |
| Kapitel 11: Codepflege | 281 |
| Teil IV: Der Top-Ten-Teil | 299 |
| Kapitel 12: Zehn weitere Tools zum Ausprobieren | 301 |
| Kapitel 13: Zehn KI-Coding-Ressourcen | 311 |
| Abbildungsverzeichnis | 319 |
| Stichwortverzeichnis | 327 |



Inhaltsverzeichnis

| | |
|--|-----------|
| Einleitung | 19 |
| Über dieses Buch | 20 |
| Törichte Annahmen über die Leser | 21 |
| Symbole, die in diesem Buch verwendet werden | 22 |
| Über dieses Buch hinaus | 23 |
| Wie es weitergeht | 23 |
| | |
| TEIL I | |
| TECHNIKEN UND TECHNOLOGIEN | 25 |
| | |
| Kapitel 1 | |
| Wie das Coding von KI profitiert | 27 |
| Langweilige Aufgaben abschieben | 27 |
| Langweilige Aufgaben erkennen | 28 |
| Die Vorlage von KI schreiben lassen | 28 |
| CRUD mit KI ausarbeiten | 31 |
| Hilfe bei der Syntax | 36 |
| Hören Sie auf, sich triviale Details merken zu wollen | 36 |
| Hinweise, wie es im Code weitergeht | 37 |
| Unterstützung bei der Syntax | 38 |
| Linting mit KI | 40 |
| Fehlerhaften Code mit statischer Codeanalyse erkennen | 40 |
| Integration von KI in die statische Codeanalyse | 41 |
| Der Einsatz von KI als Tutor | 42 |
| Das Potenzial von KI beim Lernen | 42 |
| Potenzielle Fallstricke vermeiden | 42 |
| Paar-Programmierung mit KI | 42 |
| Überblick über verschiedene Arten der Paar-Programmierung | 43 |
| Die Vor- und Nachteile der Paar-Programmierung mit KI verstehen | 43 |
| KI-Paar-Programmierungssitzung | 44 |
| | |
| Kapitel 2 | |
| Maschinelles Lernen und Deep Learning | 53 |
| Maschinelles Lernen und Deep Learning – die Grundlagen | 53 |
| Wichtige Begriffe | 54 |
| Neuronale Netze | 55 |
| Modelle trainieren und testen | 56 |
| Die Verarbeitung der natürlichen Sprache – ein Blick hinter die Kulissen | 58 |
| Geschichte der NLP | 59 |
| Die Herausforderungen des NLP | 60 |
| Transformer verstehen | 63 |
| Lernen, aufmerksam zu sein | 63 |
| Token verwenden | 64 |

14 Inhaltsverzeichnis

| | |
|--|-----------|
| Generative KI-Modelle. | 68 |
| Die Grenzen von KI erkennen | 69 |
| Sprachmodelle sind schlecht in Mathe. | 69 |
| Sprachmodelle sind geschwätzig. | 70 |
| KI hat begrenztes Wissen | 70 |
| KI fehlt der gesunde Menschenverstand | 70 |
| KI hat Probleme mit der Genauigkeit | 70 |
| KI hat das Potenzial, voreingenommen zu sein. | 71 |
| Kapitel 3 | |
| KI-Coding-Tools | 73 |
| GitHub Copilot navigieren | 73 |
| Das Copilot-Plug-in installieren | 73 |
| Effizient arbeiten mit Copilot | 75 |
| Tabnine | 80 |
| Tabnine installieren | 81 |
| Tabnine einrichten | 82 |
| Die KI-gesteuerte Code-Vervollständigung von Tabnine | 83 |
| Replit | 84 |
| Eine Website mit Replit erstellen | 86 |
| Der Replit-Arbeitsbereich | 88 |
| Kopplung mit Replit AI. | 90 |
| Kapitel 4 | |
| Coding mit Chatbots | 97 |
| Verbessern Sie Ihre Prompts! | 97 |
| Die Temperatur einstellen | 98 |
| Die Elemente eines Prompts | 100 |
| Offene und geschlossene Prompts | 101 |
| Verschiedene Arten von Prompts verwenden | 101 |
| Prompten wie ein Profi | 103 |
| Chatten mit Copilot | 104 |
| Die Slash-Befehle | 105 |
| Die Agenten von Copilot | 105 |
| Den Copilot-Chat optimal nutzen | 106 |
| Chatten mit ChatGPT. | 108 |
| Anmeldung und Einrichtung. | 108 |
| Benutzerdefinierte Anweisungen einstellen | 109 |
| In die OpenAI-Plattform eintauchen. | 114 |
| Überprüfung Ihres Guthabens. | 115 |
| Herumtollen auf dem Spielplatz | 116 |
| Ausführbare Beispiele | 117 |
| Rollen spielen | 118 |
| Einstellungen des Modells anpassen | 118 |
| Einen API-Schlüssel erhalten | 120 |
| Einen Chatbot mit OpenAI entwickeln | 121 |

TEIL II KI FÜRS CODING EINSETZEN 127

Kapitel 5 Der Weg vom Plan zum Prototyp 129

| | |
|---|-----|
| Die Projektanforderungen verstehen..... | 129 |
| Die Software-Anforderungen bestimmen | 130 |
| Fachliche Anforderungen | 130 |
| Funktionale Anforderungen | 130 |
| Nicht funktionale Anforderungen | 132 |
| Eine SRS schreiben..... | 133 |
| Code aus einer SRS erstellen..... | 136 |
| Verwendung eines Zero-Shot-Ansatzes | 136 |
| Das Problem aufschlüsseln | 137 |
| Verschmelzung von manuell geschriebenem und KI-generiertem Code | 138 |
| Den Prompt schreiben | 138 |
| Den Server schreiben | 139 |
| Folge-Prompts senden | 142 |
| Den Server testen..... | 143 |
| Few-Shot-Prompting auf dem Server implementieren..... | 144 |
| Verbesserung des Clients | 147 |
| Verlagerung der Logik der KI auf den Client | 150 |
| Tipps und Tricks für die Codegenerierung..... | 152 |
| Hören Sie nicht mit dem Programmieren auf..... | 153 |
| Seien Sie konkret | 153 |
| Denken Sie in kleinen Schritten | 153 |
| Stellen Sie Folge-Prompts | 153 |
| Lesen Sie in der offiziellen Dokumentation nach | 154 |
| Nutzen Sie Beispiele und Kontext | 154 |
| Sicherheit hat Priorität | 154 |
| Lernen Sie weiter | 154 |
| Halten Sie Ihre Tools auf dem neuesten Stand | 154 |
| Achten Sie auf die Grenzen von KI..... | 155 |

Kapitel 6 Code formatieren und verbessern 157

| | |
|---|-----|
| KI-Tools für die Code-Formatierung | 157 |
| Formatierungswerkzeuge einrichten | 158 |
| Mit Prettier die Code-Formatierung automatisieren | 160 |
| Refactoring mit KI | 165 |
| Code Smells erkennen..... | 165 |
| Code Smells mit Copilot erkennen | 169 |
| Sicheres Refactoring | 171 |
| Refactoring-Vorschläge generieren..... | 172 |
| Event-Listener richtig setzen | 172 |
| Entfernen der magischen Zahl..... | 173 |
| Verringerung der globalen Daten | 174 |

16 Inhaltsverzeichnis

| | |
|---|-----|
| Lange Funktionen korrigieren | 175 |
| Inkonsistente Benennungen korrigieren | 176 |
| Fehlende Kommentare | 177 |
| Ein besseres Tic-Tac-Toe-Spiel mit KI | 178 |

Kapitel 7

Fehler finden und beseitigen **183**

| | |
|--|-----|
| Das Ungeziefer kennen | 183 |
| Strategien zur Erkennung von Fehlern | 184 |
| Identifizierung häufiger Bugs | 184 |
| KI-gestützte Fehlerberichte mit Jam | 186 |
| Vorbeugung von Bugs mit Linting | 189 |
| Integration eines Linters | 190 |
| Installation der ESLint-Erweiterung | 191 |
| Code mit einem Linter korrigieren | 192 |
| Kombination von Linting und KI | 193 |
| Änderung der Regeln | 196 |
| Bugs mit KI aufspüren | 198 |
| Automatisierte Fehlerbehebung mit KI | 202 |
| Snyk stellt sich vor | 202 |
| Automatische Fehlerbehebungen | 205 |
| Überprüfung der automatischen Fehlerbehebung | 206 |
| Wissen, wann man automatisieren sollte | 207 |

Kapitel 8

Code übersetzen und optimieren **211**

| | |
|---|-----|
| Code in andere Sprachen übersetzen | 211 |
| Den Code auf die Übersetzung vorbereiten | 212 |
| Übersetzungsstrategien | 213 |
| Übersetzen eines kompletten Programms mit GPT-4 | 216 |
| Überprüfung des übersetzten Codes | 219 |
| Code mit KI optimieren | 220 |
| Vorschläge zur Code-Optimierung erhalten | 220 |
| Vermeiden einer vorzeitigen Optimierung | 224 |

TEIL III

CODE TESTEN, DOKUMENTIEREN UND PFLEGEN **227**

Kapitel 9

Code testen **229**

| | |
|---|-----|
| Einen Testplan schreiben | 230 |
| Entscheidung zwischen formal und agil | 230 |
| Schritt für Schritt durch den Testplanungsprozess | 230 |
| Die Rolle der KI bei der Testplanung | 231 |
| Kernfunktionalitäten identifizieren | 232 |
| Generierung von Testszenarien | 233 |

Testing Frameworks 234
 Installation von Jest 235
 Jest ausführen 237
 Testfälle erzeugen 238
 Einen Bericht lesen 241
 Analyse der Testergebnisse 242
 Testgeleitete Entwicklung mit KI 246

Kapitel 10
Code dokumentieren 253

Mit Dokumentations-Bots arbeiten 253
 Bauen Sie Ihren eigenen Dokumentationsbot 254
 Den Dokumentationsassistenten prüfen 257
 Code-Kommentare und -Anmerkungen 258
 Mintlify Doc Writer installieren und testen 259
 Underscore 261
 Visuelle Dokumentation erstellen 264
 Ein Sequenzdiagramm erstellen 265
 Ein Anforderungsdiagramm erstellen 266
 Automatisierung der API-Dokumentation mit KI 269
 Eine REST-API dokumentieren 269
 Einen Chatbot für die API-Dokumentation erstellen 274

Kapitel 11
Codepflege 281

Die vier Arten der Wartung 281
 Korrigierende Software-Wartung 282
 Adaptive Software-Wartung 282
 Perfektionierende Software-Wartung 283
 Vorbeugende Software-Wartung 283
 Einsatz von KI für die Code-Wartung 283
 Verbesserung der Codequalität durch KI 284
 Technische Schulden verstehen 285
 Erste Schritte mit Code Climate 286
 Aktivieren des Testabdeckungsberichts 289
 Analyse von Code-Qualitätsmetriken 290
 KI-gestützte Verbesserungen der Codequalität 291

TEIL IV
DER TOP-TEN-TEIL 299

Kapitel 12
Zehn weitere Tools zum Ausprobieren 301

Amazon CodeWhisperer 301
 Sourcegraph Cody 302
 DeepMind AlphaCode 302
 Google Gemini 304

18 Inhaltsverzeichnis

| | |
|--|------------|
| Codeium | 304 |
| Claude | 305 |
| Microsoft IntelliCode | 306 |
| Sourcery | 307 |
| Bugasura | 307 |
| UserWay | 308 |
| Kapitel 13 | |
| Zehn KI-Coding-Ressourcen | 311 |
| KI-Ressourcen von Code.org | 311 |
| Kaggle | 312 |
| Google Dataset Search | 312 |
| edX | 314 |
| Edabit | 314 |
| StatQuest | 315 |
| AI4All Open Learning | 316 |
| Gymnasium | 316 |
| fast.ai | 317 |
| Microsoft Learn | 317 |
| Abbildungsverzeichnis | 319 |
| Stichwortverzeichnis | 327 |