

Responsible AI in agentic software development (RAIASD)

ID RAIASD Preis auf Anfrage Dauer 1 Tag

Zielgruppe

Alle Personen, die an der Verwendung von agentenbasierten KI-Tools in der Softwareentwicklung beteiligt sind

Voraussetzungen

Allgemeine Entwicklung

Kursziele

- Verschiedene Aspekte der verantwortungsvollen Al verstehen
- Wie man generative KI verantwortungsvoll in der Softwareentwicklung einsetzt
- Schnelles Engineering für optimale Ergebnisse
- Wie man generative KI im gesamten SDLC einsetzt
- Die Herausforderungen bei der Verwendung von agentischem Genal

Kursinhalt

Eine kurze Geschichte der Künstlichen Intelligenz

- Die Ursprünge der KI
- · Neuronale Netze und "Wahrscheinlichkeitsmaschinen"
- Frühe ML-Codierungstools
- Die KI-Codierrevolution der 2020er Jahre

Verantwortungsvolle Al

- Was ist verantwortungsvolle KI?
- Rechenschaftspflicht und Transparenz
- Verringerung schädlicher Verzerrungen
- Gültigkeit und Zuverlässigkeit
- Demonstration Experimentieren mit Gültigkeit und Zuverlässigkeit in Copilot
- Erklärbarkeit und Interpretierbarkeit
- Sicherheit, Schutz, Privatsphäre und Widerstandsfähigkeit
- Sicherheit und verantwortungsvolle KI in der Softwareentwicklung

GenAl verantwortungsvoll in der Softwareentwicklung

einsetzen

- Grundlagen der LLM-Codeerzeugung
- Grundlegende Bausteine und Konzepte
- Eingabeaufforderung für Vorlagen
- Systemaufforderungen zur KI-gesteuerten Codierung
- Kann Kl... Ihre Produktivität steigern?
- Kann Kl... die "langweiligen Teile" übernehmen?
- Kann Al... gründlicher sein?
- Überprüfung des generierten Codes der Blackbox-Blues
- Die Gefahr von Halluzinationen
- Die Auswirkungen von GenAl auf die Programmierkenntnisse
- Einige weitere langfristige Auswirkungen der Nutzung von GenAl
- · Wo die KI-Codegenerierung nicht gut abschneidet
- · Schnelles Engineering
 - Warum ist ein guter Souffleur so wichtig?
 - Schaffung des Kontexts für generative KI
 - Null-Schuss-, Ein-Schuss- und Wenig-Schuss-Eingabeaufforderung
 - Vernunftbasiertes Prompt-Engineering, Gedankenkette
 - Demonstration Experimentieren mit Eingabeaufforderungen in Copilot
 - Durchsetzung und Einhaltung von Token-Limits
 - Aufforderungsmuster
 - Prompt-Muster und Prompt-Priming
 - Die 6 Kategorien von Aufforderungsmustern
 - · Einige weitere Souffleur-Ansätze
 - Least-to-Most und Self-Planning: Zerlegung komplexer Aufgaben
 - Demonstration Aufgabenzerlegung mit Copilot
 - Einheitstests, TDD und GenAl
 - Demonstration Testbasierte
 Codegenerierung mit Copilot
- Integration von generativer KI in den SDLC
 - Einsatz von GenAl über die Codegenerierung hinaus
 - Einsatz von AI bei der Anforderungsspezifikation
 - Aufforderungsmuster f
 ür die Erfassung von Anforderungen
 - Prompt-Muster für den Softwareentwurf
 - Demonstration Anforderungserfassung und API-Design mit Copilot
 - · Einsatz von Al bei der Umsetzung

Responsible AI in agentic software development (RAIASD)

- Prompt-Muster für die Umsetzung
- Demonstration Auffinden versteckter Annahmen mit Copilot
- · Einsatz von AI bei Tests und QA
- Agentische Software-Entwicklung
 - o Intelligente Agenten und GenAl
 - Wie unterscheidet sich die agenturische Kodierung?
 - Das Modell-Kontext-Protokoll (MCP)
 - Fähigkeiten von MCP-Agenten
 - Agentische Integration in IDEs
 - · Agentischer Entwicklungsablauf
 - Code-to-Spec und Spec-to-Code mit GenAl
 - Automatisierter Gerüstbau
 - Demonstration Agentisches Gerüst mit Copilot
 - Einrichten der Laufzeitumgebung
 - Demonstration Einrichtung der Umgebung mit Copilot
 - Inkrementelle Entwicklung
 - Demonstration Inkrementelle Entwicklung mit Copilot
 - Die Rolle von MCP in Dev(Sec)Ops
 - Demonstration Einsatz von MCP in DevOps mit Copilot
 - Fallstricke und bewährte Verfahren
 - "Vibe Coding" und seine Auswirkungen
 - Technische Probleme mit MCP
 - Sicherheitsaspekte der agentengestützten Entwicklung
 - Die Auswirkungen von MCP auf die Angriffsfläche
 - MCP-spezifische Angriffsvektoren
 - Demonstration Angriff auf agentischen Copiloten
 - Fallstudie Datenbankleck über Supabase MCP
 - Halluzinationen und "agentische Todesspiralen
 - Tokengrenzen und Kontext
 - Kontextverschlechterung bei sehr grosser Tokenanzahl
 - Prompt-Engineering vs. Kontext-Engineering
 - Context Engineering aus der Sicht eines Entwicklers
 - Beispiele für Kontextdokumente

Zusammenfassung und Schlussfolgerungen

- Verantwortungsvolle KI-Prinzipien in der Softwareentwicklung
- Generative AI Ressourcen und zusätzliche Anleitungen

Responsible AI in agentic software development (RAIASD)

Weltweite Trainingscenter





Fast Lane Institute for Knowledge Transfer (Switzerland) AG

Husacherstrasse 3 CH-8304 Wallisellen Tel. +41 44 832 50 80

info@flane.ch, https://www.flane.ch