



ICT Training Center

Il tuo partner per la Formazione e la Trasformazione digitale della tua azienda



SPRING AI

GENERATIVE ARTIFICIAL INTELLIGENCE CON JAVA

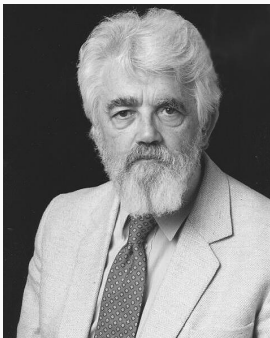
Simone Scannapieco

Corso base per Venis S.p.A, Venezia, Italia

Settembre 2025

ARTIFICIAL INTELLIGENCE

PARADIGMI



John McCarthy
@Naukas

D: Cosa è l'Intelligenza Artificiale?

R: E' la scienza e l'ingegneria di creare macchine intelligenti, in particolare programmi informatici intelligenti. È correlata al compito simile di utilizzare i computer per comprendere l'intelligenza umana, ma l'intelligenza artificiale non deve limitarsi a metodi che siano osservabili biologicamente.

John McCarthy, **Stanford Uni, 2007**

Traduzione: ©ChatGPT

- ➔ Sistema di AI è un termine estremamente inflazionato...
- ➔ ...anche per sistemi che AI non sono affatto



Questo significa che sei, ehm...in ritardo di due secoli per cena! Forse potrei prepararti uno spuntino? Devi essere affamato!

Codsworth all'Unico Sopravvissuto, **Fallout 4**, 2015

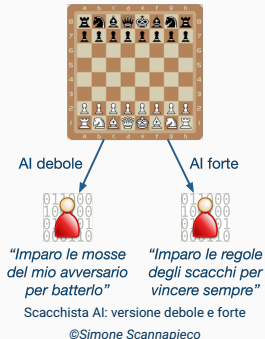
©Nukapedia: The Fallout Wiki

➡ **Non-Playable Characters (NPC)** dei videogame

⚠️ Sistemi di regole *if-then-else*, **non AI**

AI: DEBOLE VS FORTE

- ➔ **AI Debole (Narrow AI):** sistemi specializzati per compiti specifici
 - ➔ Contesto **noto a priori**, definito dai dati su cui il sistema è addestrato
 - ➔ Incapaci di generalizzare oltre il loro scopo
- ➔ **AI Forte (General AI):** sistemi autonomi e coscienti in grado di affrontare situazioni anche impreviste
 - ➔ Contesto **dinamico**, definito dalla realtà, interpretato dal sistema
 - ➔ Soluzione trovata a fronte di una **comprensione della natura del problema**
 - ➔ Creatività e intuizione
- ➔ **Super AI:** intelligenza che supera quella umana in tutti i domini
 - ➔ Teoria dell'Auton
- ⚠ Oggi esistono **solo** sistemi di AI Debole
 - ➔ Stime GAI: 2040–2060



- ➔ **AI Debole (Narrow AI):** sistemi specializzati per compiti specifici
 - ➔ Contesto **noto a priori**, definito dai dati su cui il sistema è addestrato
 - ➔ Incapaci di generalizzare oltre il loro scopo
- ➔ **AI Forte (General AI):** sistemi autonomi e coscienti in grado di affrontare situazioni anche impreviste
 - ➔ Contesto **dinamico**, definito dalla realtà, interpretato dal sistema
 - ➔ Soluzione trovata a fronte di una **comprensione della natura del problema**
 - ➔ Creatività e intuizione
- ➔ **Super AI:** intelligenza che supera quella umana in tutti i domini
 - ➔ Teoria dell'Auton
- ⚠ Oggi esistono **solo** sistemi di AI Debole
 - ➔ Stime GAI: 2040–2060



Annalee Call
Al programmata da sistemi di AI forte
©Alien: Resurrection

➔ Analisi del *sentiment*: classificare recensioni dei clienti

Analizzatore *sentiment* (© Claude)

```
@RestController
public class SentimentController {

    @Autowired
    private SentimentAnalysisService service;

    @PostMapping("/analyze")
    public SentimentResult analyze(@RequestBody String text) {
        return service.analyzeSentiment(text);
    }
}
```

➔ Chatbot per supporto: rispondere a domande frequenti

Chatbot tradizionale con regole fisse (© Claude)

```
@Component
public class ChatbotService {

    public String processQuery(String userQuery) {
        if (userQuery.contains("password")) {
            return "Per resettare la password...";
        }
        // Pattern matching limitato
        return "Non ho capito la domanda";
    }
}
```

➔ Rilevamento frodi: identificare transazioni sospette

Sistema di rilevamento frodi basato su ML (© Claude)

```
@Service
public class FraudDetectionService {

    public boolean isFraudulent(Transaction tx) {
        // Analizza pattern: importo, orario,
        // geolocalizzazione, frequenza
        return mlModel.predict(tx.getFeatures()) > 0.8;
    }
}
```

➔ Rilevamento frodi: identificare transazioni sospette

Pricing dinamico basato su AI (© Claude)

```
@RestController
public class PricingController {

    @Autowired
    private DynamicPricingService pricingService;

    @GetMapping("/price/{productId}")
    public PriceResponse getPrice(@PathVariable Long productId,
                                @RequestParam String customerSegment) {
        return pricingService.calculateOptimalPrice(productId,
                                                    customerSegment);
    }
}
```

- ➡ **Programmatore universale:** scrivere codice in qualsiasi linguaggio e dominio
- ➡ **Problem solver generale:** risolvere problemi mai visti prima
- ➡ **Apprendimento rapido:** imparare nuovi concetti da pochi esempi
- ➡ **Ragionamento causale:** comprendere cause ed effetti complessi

Analizzerò il tuo sistema legacy in COBOL, lo convertirò in microservizi Spring Boot, ottimizzerò le performance del database, implementerò la sicurezza OAuth2 e creerò una UI React responsive. Tutto completato in 30 minuti.

Scenario ipotetico di AGI programmatore

- ➡ **Programmatore universale:** scrivere codice in qualsiasi linguaggio e dominio
- ➡ **Problem solver generale:** risolvere problemi mai visti prima
- ➡ **Apprendimento rapido:** imparare nuovi concetti da pochi esempi
- ➡ **Ragionamento causale:** comprendere cause ed effetti complessi

*Il tuo e-commerce ha un calo delle vendite del 23%.
Dopo aver analizzato dati di mercato, comportamento
utenti e tendenze sociali, ho identificato 12 fattori
causali e propongo una strategia integrata che
combina UX, pricing e marketing personalizzato.*

Scenario di problem solving generale

- ➡ **Programmatore universale:** scrivere codice in qualsiasi linguaggio e dominio
- ➡ **Problem solver generale:** risolvere problemi mai visti prima
- ➡ **Apprendimento rapido:** imparare nuovi concetti da pochi esempi
- ➡ **Ragionamento causale:** comprendere cause ed effetti complessi

Non ho mai visto il framework Quantum-Spring che avete appena rilasciato, ma dopo aver letto 3 esempi di codice ho compreso i pattern architetturali e posso già ottimizzare la vostra implementazione.

Apprendimento rapido da pochi esempi

- ➡ **Programmatore universale:** scrivere codice in qualsiasi linguaggio e dominio
- ➡ **Problem solver generale:** risolvere problemi mai visti prima
- ➡ **Apprendimento rapido:** imparare nuovi concetti da pochi esempi
- ➡ **Ragionamento causale:** comprendere cause ed effetti complessi

Il bug nel vostro sistema non è dove pensate. Il vero problema è un race condition causato da un pattern di design che crea dipendenze cicliche tra tre microservizi, amplificato dal garbage collector di JVM sotto carico elevato.

Ragionamento causale profondo

CONFRONTO: AI DEBOLE VS AI FORTE

DIFFERENZE PRATICHE PER LO SVILUPPATORE

Aspetto	AI Debole (Narrow)	AI Forte (General)
Scopo	Un singolo compito specifico	Qualsiasi compito cognitivo
Apprendimento	Dataset grandi, training lungo	Pochi esempi, apprendimento rapido
Trasferibilità	Zero (limitata al dominio)	Totale (tra qualsiasi dominio)
Creatività	Assente o molto limitata	Creatività genuina
Ragionamento	Pattern matching avanzato	Comprensione causale
Implementazione	Disponibile oggi	Non esiste ancora
Costo/Complessità	Gestibile per singoli casi d'uso	Teoricamente enorme