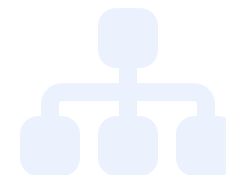


# النظم الخبيرة

## من المفهوم إلى التطبيق

---

# فهرس الموضوع



## المقدمة والنشأة

تعريف النظم الخبيرة، المشكلة التي حلتها، التاريخ والنشأة

١

## المكونات الأساسية

قاعدة المعرفة، الذاكرة العاملة، محرك الاستدلال، آلية الشرح، واجهة المستخدم

٢

## التصميم والتطبيق

من الفكرة إلى التصميم المنطقي، تمثيل المعرفة، التنفيذ البرمجي

٣

## الأمثلة التطبيقية والكود

نظام تشخيص مشاكل الإنترنت، عرض الكود النهائي بالتفصيل

٤

## التحديات والمستقبل

التحديات الحالية والقيود، التطبيقات المستقبلية والذكاء الاصطناعي الحديث

٥



# ماهي النظم الخبيرة؟

## تعريف النظم الخبيرة

النظم الخبيرة هي برامج حاسوبية مصممة لمحاكاة قدرة الإنسان الخبير على اتخاذ القرارات وحل المشكلات في مجال محدد، من خلال تمثيل المعرفة البشرية النادرة والمتخصصة وتطبيق منطق الاستدلال.

## سبب التسمية

سُميت بـ "النظم الخبيرة" لأنها:

- تعمل بطريقة مشابهة للخبير البشري في مجال معين
- تلتقط وتحفظ المعرفة المتخصصة التي يمتلكها الخبراء
- تستخدم استراتيجيات مماثلة للتفكير المنطقي البشري
- تقدم تفسيرات واستنتاجات ذات مصداقية كالخبير البشري

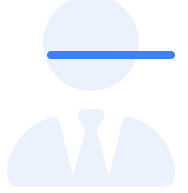
النظم الخبيرة: من المفهوم إلى التطبيق



### محاكاة التفكير الخبير البشري

- تخزين المعرفة
- تطبيق قواعد الاستدلال
- تبرير وشرح الاستنتاجات

# مشكلة فقدان الخبرة البشرية



## قصة واقعية: "السيد صالح" والنفط

تخيل نفسك مديرًا لشركة تنقيب عن النفط في سبعينيات القرن الماضي. لديك خبير جيولوجي واحد فقط، اسمه "السيد صالح"، عبّري بكل ما للكلمة من معنى. يستطيع أن ينظر إلى بيانات المسح الجيولوجي ويحدد بدقة مذهلة: "احفروا هنا، ستجدون النفط".

## المشكلة الحقيقية

ثلاث مخاوف تؤرقك كل ليلة:

- السيد صالح يتقدم في العمر وسيقاعد قريبًا
- إذا مرض أو ذهب في إجازة، تتوقف عمليات التنقيب المكلفة
- خبرته التي استغرقت 40 عامًا لبنائها ستختفي معه

المشكلة ليست برمجية، بل هي مشكلة ندرة الخبرة البشرية. كيف يمكننا التقاط هذه الخبرة النادرة، والحفاظ عليها، وتوزيعها، وجعلها متاحة في كل وقت وفي كل مكان؟

النظم الخبيرة: من المفهوم إلى التطبيق



# تاريخ ونشأة النظم الخبيرة

## جذور النظم الخبيرة

ولدت النظم الخبيرة في الستينيات والسبعينيات من رحم أبحاث الذكاء الاصطناعي، حيث كان الهدف الرئيسي حينها هو بناء أنظمة قادرة على التفكير بطريقة منطقية ومنهجية تحاكي الخبراء البشريين في مجالات محددة.

### مرحلة البدايات (١٩٦٠-١٩٧٠)

بدأت التجارب الأولى في جامعة ستانفورد بقيادة إدوارد فاينباوم وجوشوا ليدريرج، حيث ظهر نظام DENDRAL كأول نظام خبير حقيقي لتحليل المركبات الكيميائية.

### مرحلة التطور (١٩٧٠-١٩٨٠)

شهدت هذه الفترة تطور نظام MYCIN في جامعة ستانفورد بقيادة إدوارد شورتليف، والذي كان يضاهي دقة الأطباء المتخصصين في تشخيص أمراض الدم المعدية.

### مرحلة النضج والانتشار (١٩٨٠-١٩٩٠)

انتشرت النظم الخبيرة في هذه الفترة من المختبرات إلى التطبيقات التجارية والصناعية، وظهرت الأدوات والبيئات المتخصصة لبناء النظم الخبيرة مثل CLIPS و OPS5.

## النظم الخبيرة الرائدة

### DENDRAL

١٩٦٥

أول نظام خبير لتحليل المركبات الكيميائية وتحديد بنيتها الجزيئية



### MYCIN

١٩٧٢

نظام خبير لتشخيص أمراض الدم المعدية واقتراح المضادات الحيوية المناسبة



# فرق جوهري: البرنامج التقليدي مقابل النظام الخبير



## النظام الخبير

يتعامل مع **المعرفة** (Knowledge)

- يُحاكي تفكير الخبير البشري وقواعده
- يطبق الاستدلال المنطقي على الحقائق
- يستخلص استنتاجات جديدة
- يستطيع شرح وتبرير الاستنتاجات

### مثال: التشخيص الطبي

يستقبل أعراضاً (حرارة، سعال) ويطبق قواعد استنتاجية ليقتراح تشخيصاً (انفلونزا) مع شرح السبب.

VS



## البرنامج التقليدي

يتعامل مع **البيانات** (Data)

- يعالج أرقاماً ونصوصاً بطريقة محددة
- يتبع خطوات ثابتة ومبرمجة سلفاً
- يستخدم خوارزميات حسابية مباشرة
- لا يشرح سبب التوصل للنتيجة

### مثال: الآلة الحاسبة

تستقبل مدخلات (3+5) وتعالجها وفق خوارزمية محددة سلفاً لتعطي نتيجة (8) بلا تفسير.

## الفرق الجوهري:

في البرامج التقليدية، **"المعرفة"** (المنطق) تكون مدمجة ومتشابكة مع **"الكود"** (إجراءات المعالجة). أما في النظم الخبيرة، فهناك **فصل تام** بين المعرفة (قاعدة المعرفة) والمعالجة (محرك الاستدلال)، مما يسمح بتعديل المعرفة بسهولة دون تغيير الكود.

# خريطة المكونات الأساسية للنظام الخبير

يتكون النظام الخبير من خمسة مكونات رئيسية متكاملة تعمل معًا لمحاكاة عملية التفكير واتخاذ القرار لدى الخبير البشري:



## قاعدة المعرفة

مستودع القواعد والخبرات المتخصصة



## الذاكرة العاملة

مخزن الحقائق والمعطيات المؤقتة



## محرك الاستدلال

قلب النظام الخبير ومحرك التفكير المنطقي



## واجهة المستخدم

نقطة التواصل بين المستخدم والنظام

### التكامل بين المكونات:

تعمل هذه المكونات معًا في تناغم تام: **قاعدة المعرفة** توفر القواعد والخبرة، بينما تخزن **الذاكرة العاملة** معطيات الحالة الراهنة. يقوم **محرك الاستدلال** بتطبيق القواعد على الوصول لاستنتاجات، وتشرح **آلية الشرح** هذه الاستنتاجات للمستخدم عبر **واجهة المستخدم** سهلة الاستخدام.

# قاعدة المعرفة (Knowledge Base): حجر الأساس

## تعريف قاعدة المعرفة

قاعدة المعرفة هي المستودع الذي يخزن جميع المعارف المتخصصة والخبرات اللازمة لحل المشكلات في مجال معين. إنها تمثل "حكمة الخبير" في شكل منظم وقابل للاستخدام من قبل الحاسوب.

## وظيفة قاعدة المعرفة

- تخزين الخبرة المتخصصة بطريقة منطقية ومنظمة
- تمثيل العلاقات السببية بين الأحداث والظواهر
- توفير "قواعد ذهبية" لحل المشكلات بناءً على أنماط معينة
- الفصل بين المعرفة والآليات التي تستخدمها (منطق العمل)
- تمكين تحديث وتطوير المعرفة دون تغيير هيكل النظام

## التشبيه الأقرب

يمكن تشبيه قاعدة المعرفة بـ "كتاب وصفات" خاص بالشيف الماهر، حيث:

- الصفات تمثل "القواعد" لحل مشكلات محددة
- المكونات المطلوبة تمثل "الشروط" اللازمة
- الطبق النهائي يمثل "الاستنتاج" أو الحل



### كتاب القواعد الذهبية للخبير

#### أمثلة لقواعد المعرفة:

قاعدة #1:

إذا (لا تعمل السيارة) و (الأضواء ضعيفة)  
إذن (المشكلة في البطارية)

قاعدة #2:

إذا (المشكلة في البطارية) و (وجود تآكل)  
إذن (تنظيف أقطاب البطارية)



# تمثيل المعرفة بقواعد IF-THEN

## نموذج القواعد الإنتاجية

القواعد الإنتاجية (Production Rules) هي أكثر طرق تمثيل المعرفة شيوعًا في النظم الخبيرة، وتأخذ الشكل العام:

**IF (شرط أو مجموعة شروط) THEN (استنتاج)**

### مثال من واقع تشخيص أعطال السيارات:

- القاعدة #1: إذا كانت السيارة لا تعمل **و** الأضواء ضعيفة **إذن** المشكلة في البطارية.
- القاعدة #7: إذا كانت المشكلة في البطارية **و** يوجد تآكل على الأقطاب **إذن** الحل هو تنظيف أقطاب البطارية.

### فوائد قواعد IF-THEN:

- تمثيل المعرفة بشكل طبيعي وقريب من تفكير الخبير
- إمكانية التعديل والإضافة دون تغيير بنية النظام

# تفصيل: قاعدة المعرفة VS قاعدة البيانات



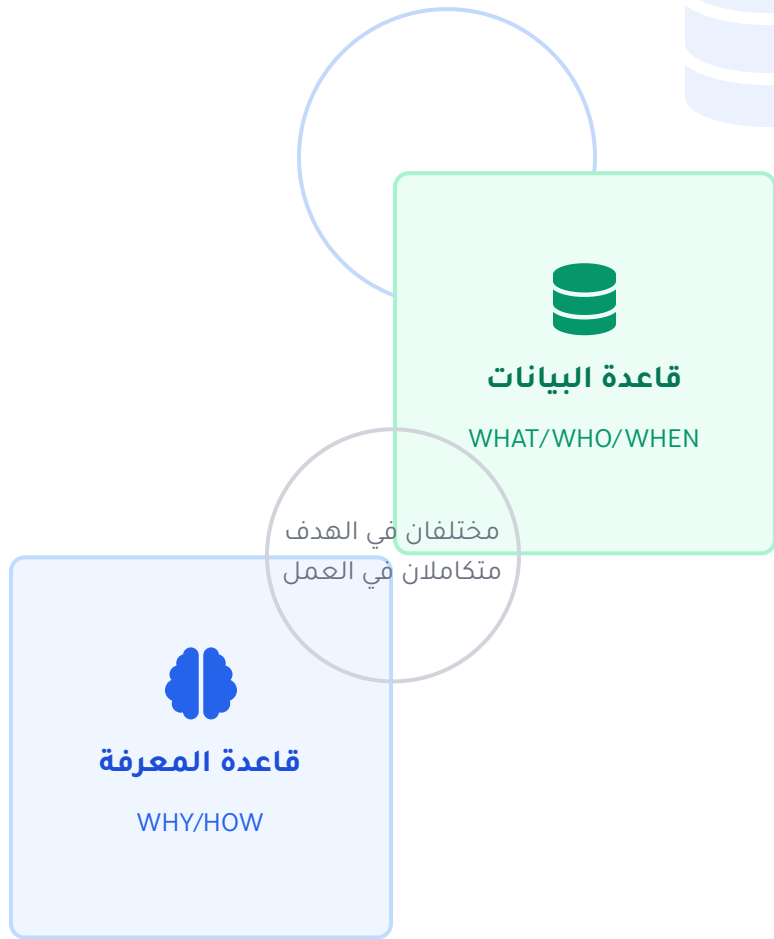
## المقارنة العميقة

معيار المقارنة	قاعدة المعرفة	قاعدة البيانات
المحتوى	استراتيجيات، قواعد، خبرات	حقائق، سجلات، بيانات خام
الهدف	التفكير، الاستدلال، حل المشكلات	تخزين واسترجاع المعلومات
الهيكلية	مرنة (قواعد IF-THEN)	صارمة (جداول، علاقات)
الاستعلام	استنتاج وتبرير (Inference)	استرجاع (Query)

أهمية الفصل بين النوعين:



- قاعدة المعرفة تركز على **لماذا وكيف** نتخذ القرار
- قاعدة البيانات تركز على **ماذا ومتى ومن**
- تسمح قاعدة المعرفة بتعديل استراتيجيات التفكير دون تغيير الكود
- تتيح النظم الخبرة الاستخدام المرن للمعرفة لحل مشكلات معقدة



# الذاكرة العاملة (Working Memory)



## ما هي الذاكرة العاملة؟

الذاكرة العاملة (Working Memory) هي المكوّن المؤقت والديناميكي في النظام الخبير. إنها تشبه "سبورة" أو "ورقة ملاحظات" مؤقتة تُستخدم لتسجيل الحقائق المتعلقة بالمشكلة الحالية فقط.



## تشبيه سبورة الطاولة

تخيل أن الذاكرة العاملة كـ:

- سبورة مؤقتة يستخدمها الميكانيكي لتدوين أعراض السيارة المعطلة.
- في بداية الفحص، تكون السبورة فارغة تماماً.
- أثناء الفحص، يضيف الميكانيكي ملاحظاته: "السيارة لا تعمل"، "أضواء الطبلون ضعيفة".
- يستنتج الميكانيكي حقائق جديدة ويضيفها: "المشكلة هي البطارية".
- عند انتهاء المهمة، يمسح السبورة استعداداً للسيارة التالية.



## الفرق الجوهرى:

قاعدة المعرفة ثابتة ودائمة (تمثل خبرة المتخصص على المدى الطويل)، بينما الذاكرة العاملة متغيرة ومؤقتة (تمثل معرفة النظام بالمشكلة الحالية فقط).

## سبورة الميكانيكي المؤقتة

حقيقة 1: السيارة لا تعمل

حقيقة 2: أضواء الطبلون ضعيفة

حقيقة مستنتجة: المشكلة هي البطارية



# محرك الاستدلال (Inference Engine): قلب التفكير المنطقي

## تعريف محرك الاستدلال

محرك الاستدلال هو "عقل" النظام الخبير، المسؤول عن تطبيق المنطق وقواعد الاستدلال على الحقائق المتوفرة في الذاكرة العاملة. يتميز بأنه منفصل تماماً عن المعرفة نفسها، ويركز فقط على آليات تطبيق المنطق.

## أنواع الاستدلال الرئيسية

### الاستدلال الخلفي (Backward Chaining) →

- يبدأ من **الهدف المفترض** ويبحث عن الدليل
- مدفوع بالهدف (Goal-Driven)
- مناسب لمشكلات **التحقق والإثبات**
- مثال: "هل المريض مصاب بالإنفلونزا؟ ما الأدلة؟"

### الاستدلال الأمامي (Forward Chaining) ←

- يبدأ من **الحقائق المتوفرة** للوصول إلى الاستنتاج
- مدفوع بالبيانات (Data-Driven)
- مناسب لمشكلات **التشخيص** (أعطال السيارات، الأمراض)
- مثال: "لدي هذه الأعراض، ما هو المرض؟"



### العقل المفكر للنظام

#### دورة التعرف والتصرف ⚙️

١

مطابقة القواعد مع الحقائق



٢

حل التعارض بين القواعد



٣

تنفيذ القاعدة المختارة

# تفصيل دورة العمل: الاستدلال الأمامي (Forward Chaining)

## كيف يعمل الاستدلال الأمامي؟

الاستدلال الأمامي هو استراتيجية "مدفوعة بالبيانات" (Data-Driven) تبدأ من الحقائق الأولية وتعمل نحو الهدف. المحرك يكرر دورة المطابقة والتنفيذ حتى لا توجد قواعد جديدة يمكن تفعيلها.

## مثال تطبيقي

لنفترض أن لدينا:

الحقائق الأولية:

القواعد:

R1: IF F1&F2 THEN F4:  
المشكلة\_هي\_البطارية  
R15: IF F4&F3 THEN F5:  
الحل\_هو\_تنظيف\_الأقطاب

F1: السيارة\_لا\_تعمل  
F2: أعضاء\_الطبلون\_لا\_تعمل  
F3: يوجد\_تآكل\_على\_الأقطاب

## تطبيق دورة العمل:

### الدورة الأولى:

- المطابقة: فحص R1 (شروطه F1 و F2 موجودة)
- حل التعارض: اختيار R1
- التنفيذ: إضافة F4 (المشكلة\_هي\_البطارية) للذاكرة العاملة

### الدورة الثانية:

- المطابقة: فحص R15 (شروطه F4 و F3 موجودة الآن)
- حل التعارض: اختيار R15
- التنفيذ: إضافة F5 (الحل\_هو\_تنظيف\_الأقطاب) للذاكرة العاملة

### الدورة الثالثة:

- المطابقة: لا توجد قواعد أخرى يمكن تفعيلها
- النتيجة النهائية: توقف المحرك وعرض الاستنتاجات

## دورة التعرف-التصرف (Recognize-Act Cycle)



# آلية الشرح (Explanation Facility)

## ما هي آلية الشرح؟

آلية الشرح هي مكون في النظام الخبير مسؤول عن تفسير سلوك النظام واستنتاجاته للمستخدم. إنها قدرة النظام على إظهار "سبب" اتخاذها لقرار معين أو وصوله إلى استنتاج محدد.

## أهميتها في بناء الثقة

- بدون تفسيرات، يظل النظام "صندوقاً أسود" غامضاً
- المستخدم لن يثق بالنظام إذا لم يفهم كيفية وصوله للاستنتاجات
- تمكّن المستخدم من التحقق من صحة المنطق المستخدم
- تسمح بتصحيح الأخطاء في قاعدة المعرفة

## أنواع الأسئلة التي تجيب عليها

### كيف؟ (How?)

توضح سلسلة الاستدلال التي أدت إلى نتيجة أو استنتاج معين

### لماذا؟ (Why?)

تشرح سبب طرح النظام لسؤال معين أو طلبه لمعلومة محددة



### بناء الثقة من خلال الحوار المنطقي

لماذا تعتقد أن المشكلة في البطارية؟

لأن السيارة لا تعمل وأضواء الطبلون ضعيفة.

كيف توصلت إلى هذا الحل؟

### شرح الاستدلال:

1. حقيقة: السيارة لا تعمل
2. حقيقة: أضواء الطبلون ضعيفة
3. تفعيل القاعدة: إذا السيارة لا تعمل وأضواء الطبلون ضعيفة فالمشكلة في البطارية

# آلية الشرح: تطبيق واقعي مع نظام سيارات

## بناء الثقة من خلال الشفافية

آلية الشرح تتيح للنظام الخبير أن يقدم تبريرات منطقية لاستنتاجاته، مما يبني الثقة بين المستخدم والنظام.

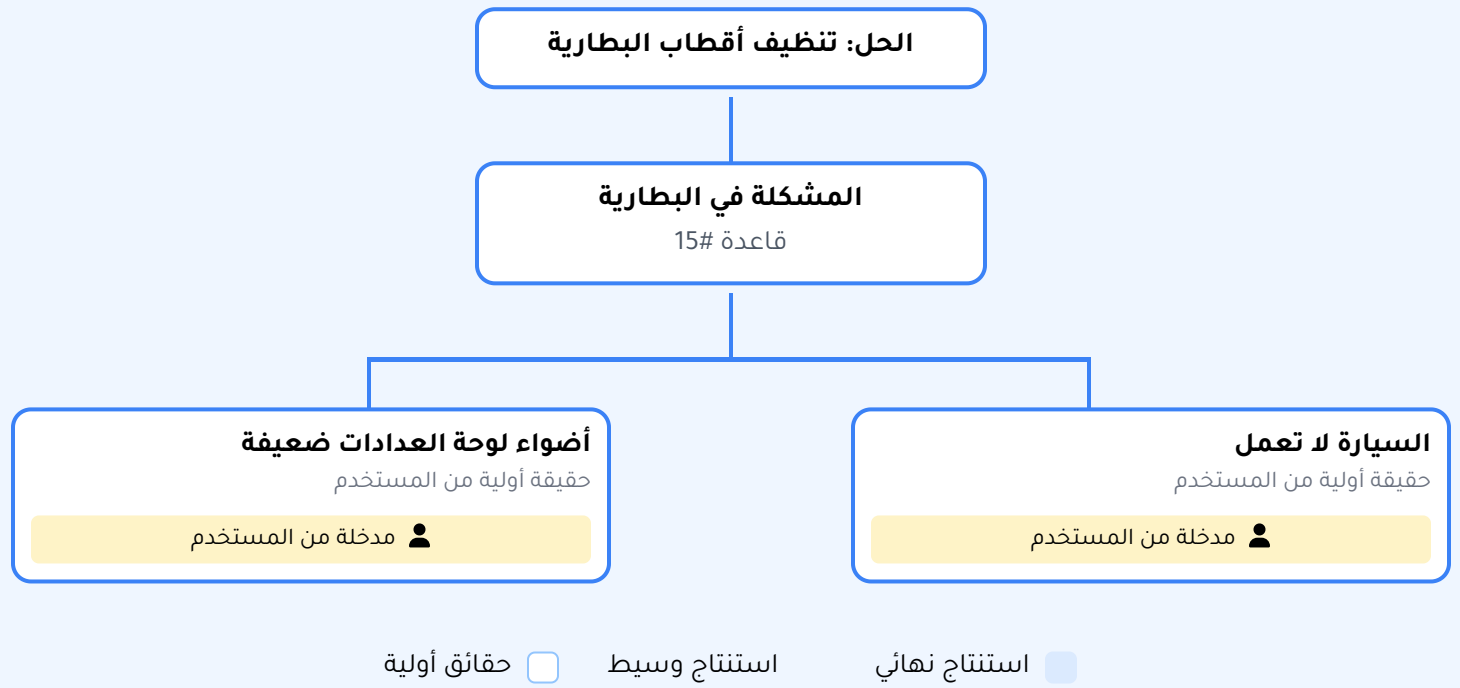
### مثال حوار حقيقي:

**النظام:** "المشكلة في البطارية."  
**المستخدم:** "كيف عرفت ذلك؟"  
**النظام:** "لقد استنتجت ذلك لأن السيارة لا تعمل وأضواء لوحة العدادات ضعيفة، وهذه أعراض تشير غالباً إلى ضعف البطارية."

### فوائد آلية الشرح:

- تزيد من ثقة المستخدم بالنظام
- تساعد في تصحيح الأخطاء وتحسين القواعد
- تتيح التعلم للمستخدمين المبتدئين

## شجرة الاستدلال المنطقي



# واجهة المستخدم (User Interface): الجسر بين الإنسان والنظام

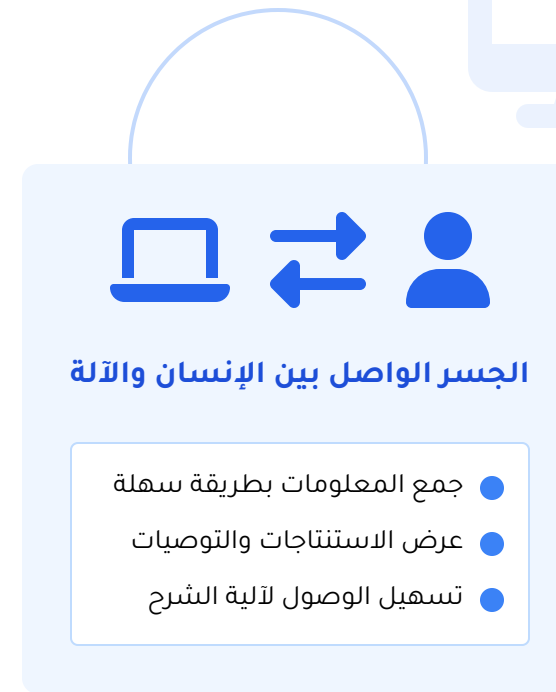
## ما هي واجهة المستخدم في النظام الخبير؟

هي الجزء المرئي والتفاعلي الذي يسمح للمستخدم بالتواصل مع النظام الخبير. تلعب دورًا حيويًا في تقديم المعلومات المعقدة بطريقة سهلة ومفهومة للمستخدم غير المتخصص في البرمجة.

## وظائف واجهة المستخدم

- جمع الحقائق الأولية من المستخدم (أسئلة، خيارات متعددة، قوائم منسدلة)
- عرض النتائج النهائية بلغة بشرية مفهومة وليس برموز برمجية
- تسهيل استخدام آلية الشرح (أزرار "لماذا؟" و "كيف؟")
- تقديم وسائل مساعدة للمستخدم (صور توضيحية، شروحات للمصطلحات)
- تخفيف "العبء المعرفي" على المستخدم وجعل التفاعل طبيعيًا

النظم الخبيرة: من المفهوم إلى التطبيق



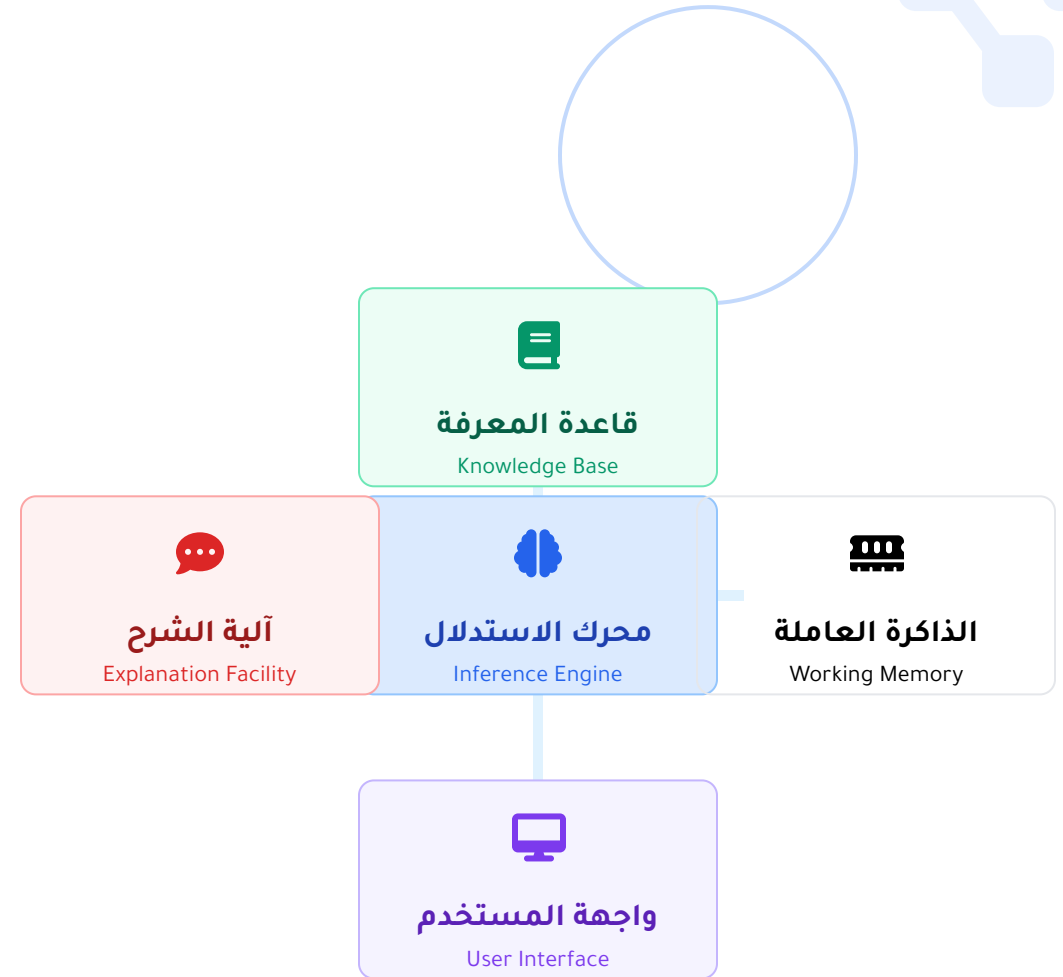


# من النموذج النظري إلى مخطط التصميم

## تجميع مكونات النظام الخبير

عند تصميم نظام خبير متكامل، يتم ربط المكونات الخمسة في منظومة واحدة متناغمة تعمل معاً بتكامل تام:

- **قاعدة المعرفة:** تزود محرك الاستدلال بالقواعد والحقائق الثابتة التي تمثل خبرة الخبير البشري.
  - **الذاكرة العاملة:** تحتفظ بالحقائق الخاصة بالمشكلة الحالية وتتحدث مع محرك الاستدلال بشكل مستمر.
  - **محرك الاستدلال:** قلب النظام المسؤول عن تطبيق قواعد المعرفة على الحقائق الحالية للوصول إلى استنتاجات.
  - **آلية الشرح:** تراقب عمل محرك الاستدلال وتسجل مسار التفكير لتبرير الاستنتاجات للمستخدم.
  - **واجهة المستخدم:** تمثل نقطة الاتصال مع المستخدم، حيث تجمع المدخلات وتعرض النتائج والشروح بطريقة مفهومة.
- تدفق البيانات بين المكونات يتبع نمطاً محدداً، حيث يبدأ من واجهة المستخدم المستخدم نحو الذاكرة العاملة، ثم يعمل محرك الاستدلال مستخدماً قاعدة المعرفة، وأخيراً تعود النتائج عبر آلية الشرح إلى المستخدم.



# تحويل الحوار البشري إلى قواعد

## كيف نحول الخبرة إلى قواعد منطقية؟

هندسة المعرفة (Knowledge Engineering) هي عملية استخلاص وتنظيم المعرفة من الخبير البشري وتحويلها إلى قواعد يمكن للحاسوب فهمها وتطبيقها.



## خطوات العملية

1. **المقابلة مع الخبير:** طرح أسئلة محددة وتسجيل الإجابات بدقة.
2. **تحديد الشروط والنتائج:** تحليل حوار الخبير لاستخراج الشروط ("إذا كان كذا") والنتائج ("إذن كذا").
3. **صياغة القواعد:** تحويل العبارات الطبيعية إلى قواعد IF-THEN رسمية.
4. **تحقق وتنقيح:** مراجعة القواعد مع الخبير للتأكد من دقتها واكتمالها.

## مثال آخر

أبو خالد: "لو الزبون قال إن السيارة تطفئ فجأة وهو ماشي، وإذا شमित ريحة بنزين، غالباً المشكلة في مضخة الوقود."

IF (السيارة\_تطفئ\_أثناء\_القيادة) AND (توجد\_رائحة\_بنزين) THEN (المشكلة\_في\_مضخة\_الوقود)

## من الحوار الطبيعي...

أبو خالد (الميكانيكي):

"إذا السيارة ما تشتغل، والأضواء ضعيفة، أول شيء أفحصه هو البطارية."



## ...إلى قواعد منطقية منظمة

القاعدة #15:

IF (شرط):

- السيارة\_لا\_تعمل = TRUE  
- أضواء\_الطبلون\_ضعيفة = TRUE

THEN (نتيجة):

- المشكلة\_المحتملة\_هي\_البطارية = TRUE

# هيكل القواعد والحقائق: تمثيل منطقي

## مكونات القواعد والحقائق

تعتمد النظم الخبيرة على التمثيل المنطقي للمعرفة من خلال هيكلين أساسيين:

### الحقائق (Facts)

- وحدة المعرفة الأساسية في النظام
- بناء بسيط يتكون من وصف نصي فقط ( حقيقة ("السيارة\_لا\_تعمل" )
- يمكن أن تكون أولوية (من المستخدم) أو مستنتجة (من القواعد)
- يجب أن تكون قابلة للمقارنة (equality) لتحديد التطابق

### القواعد (Rules)

- تتكون من جزأين رئيسيين:
  - الشروط:** قائمة من الحقائق المطلوب توفرها معاً
  - النتيجة:** حقيقة واحدة جديدة تُستنتج عند تحقق الشروط
- تمثل القواعد العلاقة المنطقية "إذا تحقق كذا... إذن كذا"
- يمكن التعبير عن العلاقات المنطقية المعقدة من خلال تكوين قواعد متعددة متكاملة

### تمثيل قاعدة منطقية في البرمجة:

```
Rule  
(  
  conditions=[Fact("السيارة_لا_تعمل"), Fact("أعضاء_الطبلون_ضعيفة")],  
  conclusion=Fact("المشكلة_هي_البطارية")  
)
```

### هيكل القاعدة (Rule)

#### الشروط (Conditions):

حقيقة 1: السيارة\_لا\_تعمل

حقيقة 2: أعضاء\_الطبلون\_ضعيفة

AND (و)

#### النتيجة (Conclusion):

حقيقة جديدة: المشكلة\_هي\_البطارية

### هيكل الحقيقة (Fact)

الوصف: "السيارة\_لا\_تعمل"

# تصميم البيانات: بناء Rule و Fact برمجياً

## تصميم الهياكل البياناتية الأساسية

لبناء النظام الخبير، نحتاج إلى تصميم هياكل البيانات بشكل يحقق:

- الفصل التام بين تمثيل المعرفة والعمليات المنطقية
- سهولة المقارنة بين الحقائق للتحقق من تطابقها
- قابلية تمثيل العلاقات بين الشروط والنتائج

### العنصران الأساسيان

#### Rule - القاعدة

رابط منطقي بين الشروط والنتائج

#### Fact - الحقيقة

وحدة المعلومات الأساسية التي لا تتجزأ

### البرمجة كائنية التوجه في النظم الخبيرة:

تقدم البرمجة كائنية التوجه (OOP) إطاراً مثالياً لتصميم النظم الخبيرة، حيث تساعد في:

● التغليف (Encapsulation): تجميع المعطيات والأساليب معاً

تساوي: تعريف كيفية مقارنة الحقائق

# محرك الاستدلال: التصميم على الورق

## وظيفة "تشغيل\_المحرك"

قبل كتابة أي كود برمجي، يجب تصميم خوارزمية واضحة لمحرك الاستدلال على الورق:

1. إنشاء حلقة تكرارية (Loop) تستمر ما دامت هناك حقائق جديدة:
2. البحث عن القواعد القابلة للتفعيل (شروطها متحققة)
3. اختيار قاعدة واحدة من القواعد المرشحة
4. تنفيذ القاعدة المختارة (إضافة استنتاجها للذاكرة)
5. التحقق من وجود قواعد جديدة للتفعيل
6. إنهاء الدورة عندما لا يوجد المزيد من القواعد

## النقاط الحرجة في التصميم:

- استراتيجية حل التعارض: كيفية اختيار قاعدة من بين عدة قواعد قابلة للتفعيل
- تجنب الدورات اللانهائية: عدم تفعيل نفس القاعدة مرتين
- كفاءة البحث: تصميم آلية بحث فعالة للمطابقة عند وجود قواعد كثيرة
- التسجيل والتتبع: حفظ سلسلة الاستنتاج لأغراض الشرح

## مخطط منطقي لآلية عمل المحرك

### المدخلات

قائمة القواعد (Rules) + مجموعة الحقائق الأولية (Facts)

### المعالجة (دورة التفعيل)

1. المطابقة: إيجاد القواعد المناسبة
2. حل التعارض: اختيار قاعدة واحدة
3. التنفيذ: إضافة استنتاجات جديدة

### المخرجات

مجموعة الحقائق النهائية + سلسلة الاستدلال

# تحويل من التصميم إلى البرمجة (Python OOP)

## البرمجة كائنية التوجه للنظم الخبيرة

نستخدم البرمجة كائنية التوجه (OOP) لبناء نظامنا الخبير لأنها:

- تسمح بتجريد المفاهيم المعقدة (الحقائق، القواعد) ككائنات
- تدعم فصل المسؤوليات (قاعدة المعرفة، محرك الاستدلال)
- توفر إمكانية التوسعة والتخصيص

### ما الذي نقوم ببنائه؟

كلاس `SimpleExpertSystem` يمثل الهيكل العام للنظام الخبير ويحتوي على:

- مخزن للقواعد (قاعدة المعرفة)
- مخزن للحقائق (الذاكرة العاملة)
- وظيفة لإضافة الحقائق والقواعد
- وظيفة لتشغيل محرك الاستدلال (run)

من هذا الهيكل العام، يمكننا لاحقاً إنشاء أنظمة متخصصة باستخدام الوراثة (Inheritance) مثل نظام تشخيص أعطال الإنترنت `InternetTroubleshooter`.

# الذاكرة العاملة وجمع الحقائق من المستخدم

## تعبئة الذاكرة العاملة بالحقائق الأولية

محرك الاستدلال يحتاج إلى "وقود أولي" ليبدأ عمله - وهو الحقائق الأساسية التي تمثل المشكلة الحالية. دالة `get_initial_facts()` هي المسؤولة عن:

- إجراء حوار تفاعلي مع المستخدم
- طرح أسئلة مباشرة لجمع المعلومات
- تحويل إجابات المستخدم إلى حقائق (Facts) وتخزينها في الذاكرة العاملة

💡 الذاكرة العاملة هي "المساحة المؤقتة" التي يتم فيها تخزين معلومات المشكلة الحالية فقط. تبدأ فارغة، وتُملأ خلال جلسة التشخيص، وتُفْرغ عند الانتهاء.

## سير العمل:

1. بداية الحوار مع المستخدم
  2. طرح سؤال محدد وتلقي الإجابة
  3. بناءً على الإجابة، إنشاء حقيقة Fact (جديدة)
  4. إضافة الحقيقة إلى `self.facts` (الذاكرة العاملة)
- إضافية حسب الإجابات السابقة

# تشغيل محرك الاستدلال مع الشرح الفوري

## كيف يعمل محرك الاستدلال مع الشرح الفوري؟

عند تشغيل محرك الاستدلال، يقوم بدورة متكررة من المطابقة والتنفيذ، مع توفير شرح فوري للمستخدم لكل خطوة:

### دورة تنفيذ المحرك:

1. البحث عن القواعد القابلة للتفعيل (المطابقة)
2. اختيار قاعدة واحدة للتنفيذ (حل التعارض)
3. تنفيذ القاعدة وإضافة استنتاجها للذاكرة العاملة
4. عرض تفسير للخطوة التي تم تنفيذها للمستخدم
5. تكرار العملية حتى لا توجد قواعد جديدة للتفعيل

### مخرجات الشرح الفوري للمستخدم:

```
=====
بدء تشغيل محرك الاستدلال...
=====
[تفعيل القاعدة]: IF Fact ("الإنترنت_لا_يعمل") AND
Fact ("المبة_الراوتر_لا_تعمل") THEN
Fact ("المشكلة_هي_كهرباء_الراوتر")
دورة مستنتجة: Fact ("المشكلة_هي_كهرباء_الراوتر")
```



# بناء التطبيق المتخصص: تشخيص مشاكل الإنترنت

## تخصيص النظام الخبير لمجال محدد

بعد بناء الإطار العام للنظام الخبير، يأتي دور تخصيصه لمجال محدد من خلال:

- الوراثة (Inheritance): نرث من الفئة العامة SimpleExpertSystem ونضيف التخصص.
- إنشاء قاعدة المعرفة: تعريف الحقائق والقواعد الخاصة بمجال مشاكل الإنترنت.
- الفصل بين المعرفة والمنطق: المنطق في الفئة الأم، والمعرفة في الفئة المتخصصة.

### الميزة الأساسية:

يمكننا إنشاء أنظمة خبيرة متخصصة متعددة (تشخيص سيارات، مشاكل إنترنت، تشخيص طبي) دون تغيير منطق المعالجة الأساسي!

### المكونات الرئيسية:

- `__init__`: استدعاء دالة البناء للفئة الأم وإعداد قاعدة المعرفة
- `build_knowledge_base`: تعريف الحقائق والقواعد المتخصصة

# من الفكرة إلى البرنامج: دمج كل الأجزاء

## تجميع الأجزاء في نظام متكامل

لإنشاء نظام خبير كامل، نحتاج إلى تجميع المكونات المختلفة بتسلسل منطقي:

### ١ البناء والإعداد

إنشاء النظام الخبير بقاعدة المعرفة المتخصصة، وتحديد القواعد والهياكل البيانية.

### ٢ جمع المعلومات الأولية

استخدام واجهة المستخدم لطرح الأسئلة وجمع الحقائق الأولية وتخزينها في الذاكرة العاملة.

### ٣ عملية الاستدلال

تشغيل محرك الاستدلال لمقارنة الحقائق بالقواعد وتوليد استنتاجات جديدة، مع تسجيل خطوات التفكير.

### ٤ عرض النتائج والتبرير

تقديم النتائج النهائية للمستخدم مع شرح كيفية الوصول إليها وسلسلة الاستدلال المنطقي.

# أمثلة تطبيقية: تشخيص أعطال الإنترنت - الحوار التفاعلي

## مكونات الحوار التفاعلي

- واجهة استقبال وترحيب بالمستخدم
- أسئلة موجهة لتغذية الذاكرة العاملة بالحقائق
- تحقق من الإجابات وقبول القيم المحددة
- عرض الاستنتاجات والحلول للمستخدم
- تقديم شرح مفصل عن سبب الوصول للحل

## مميزات الحوار

- بسيط وسهل الفهم
- يتفاعل بناءً على إجابات المستخدم
- يظهر خطوات التفكير بشفافية
- يبني الثقة من خلال التبرير المنطقي

### محاكاة حوار بين المستخدم والنظام الخبير

#### النظام الخبير:

مرحباً بك في نظام تشخيص مشاكل الإنترنت. سأساعدك في حل مشكلتك. أجب على الأسئلة التالية.

#### النظام الخبير:

هل الإنترنت لا يعمل؟ (نعم/لا)

#### المستخدم:

نعم

#### النظام الخبير:

ما هي حالة لمبة الإنترنت على الراوتر؟ (خضراء/برتقالية/لا تعمل)

#### المستخدم:

لا تعمل

#### النظام الخبير:

شكراً لك، تم جمع الحقائق الأولية.

#### تشغيل محرك الاستدلال:

[تفعيل القاعدة]: IF Fact ("الإنترنت\_لا\_يعمل") AND Fact ("لمبة\_الراوتر\_لا\_تعمل") THEN

Fact ("المشكلة\_هي\_كهرباء\_الراوتر")

[حقيقة جديدة مستنتجة]: Fact ("المشكلة\_هي\_كهرباء\_الراوتر")

#### تشغيل محرك الاستدلال:

[تفعيل القاعدة]: IF Fact ("المشكلة\_هي\_كهرباء\_الراوتر") THEN

Fact ("الحل\_هو\_فحص\_كابل\_الكهرباء\_والمقبس")

[حقيقة جديدة مستنتجة]: Fact ("الحل\_هو\_فحص\_كابل\_الكهرباء\_والمقبس")

#### النتيجة النهائية:

المشكلة هي كهرباء الراوتر

الحل هو فحص كابل الكهرباء والمقبس

#### المستخدم:

كيف توصلت إلى هذه النتيجة؟

#### الشرح:

توصلت إلى أن "الحل هو فحص كابل الكهرباء والمقبس" لأن:

1. لاحظت أن "المشكلة هي كهرباء الراوتر" (من القاعدة الأولى)

2. وصلت إلى استنتاج "المشكلة هي كهرباء الراوتر" لأن:

- ذكرت أن "الإنترنت لا يعمل"

- وأخبرتني أن "لمبة الراوتر لا تعمل"





# تشغيل النظام وعرض النتائج النهائية

## شرح مسار اتخاذ القرار

يعرض المثال تجربة حية لتشخيص مشكلة في الإنترنت باستخدام نظامنا الخبير. يمكننا متابعة مسار التفكير المنطقي للنظام:

### جمع الحقائق الأولية

الإنترنت لا يعمل + لمبة الراوتر برتقالية



### تفعيل القاعدة الأولى

IF (الإنترنت لا يعمل AND لمبة\_الراوتر\_برتقالية)  
THEN (المشكلة\_هي\_إشارة\_الشركة)



### تفعيل القاعدة الثانية

IF (المشكلة\_هي\_إشارة\_الشركة)  
THEN (الحل\_هو\_الاتصال\_بشركة\_الإنترنت)

هذا المثال يوضح فعالية آلية الشرح في النظام الخبير، حيث يمكننا متابعة  
تفكير المنطقي خطوة بخطوة، مما يزيد من ثقة المستخدم في  
النتائج والتوصيات المقدمة.

# التحديات التطبيقية والقيود للنظم الخبيرة

## القيود الحالية

- **محدودية المجال:**  
تعمل النظم الخبيرة بشكل جيد فقط في مجالات محددة ومعزولة، وتفشل في التعامل مع المشاكل المعقدة متعددة المجالات.
- **كفاءة الأداء:**  
مع ازدياد عدد القواعد، تنخفض كفاءة محرك الاستدلال وتزداد تعقيدات حل التعارضات.
- **عدم التأكد:**  
تعامل محدود مع حالات عدم التأكد والبيانات غير المكتملة مقارنة بتقنيات التعلم الآلي الحديثة.
- **عدم القدرة على التعلم:**  
معظم النظم الخبيرة لا تمتلك القدرة على التعلم من التجارب السابقة وتحسين أدائها تلقائياً.

## التطلع إلى المستقبل

دمج النظم الخبيرة مع تقنيات الذكاء الاصطناعي الحديثة لتجاوز هذه القيود

النظم الخبيرة: من المفهوم إلى التطبيق

## التحديات الرئيسية

### هندسة المعرفة (Knowledge Engineering)

صعوبة استخلاص المعرفة من الخبراء وتحويلها إلى قواعد صريحة. الخبير قد لا يستطيع شرح كيفية اتخاذه للقرارات بشكل واضح ومنطقي.

### المعرفة الضمنية (Tacit Knowledge)

عدم القدرة على تمثيل "الحدس" و"الخبرة الضمنية" للخبير التي لا يمكن صياغتها بسهولة في قواعد IF-THEN.

### الصيانة والتحديث (Maintenance)

تحديث قاعدة المعرفة عملية مكلفة وتتطلب تدخلاً مستمراً من مهندسي المعرفة والخبراء البشريين.

# مستقبل النظم الخبيرة والذكاء الاصطناعي الحديث

## تطبيقات المستقبل

### • الصحة الشخصية المخصصة:

أنظمة خبيرة متكاملة مع أجهزة قابلة للارتداء تقدم استشارات طبية مخصصة بناءً على البيانات الحيوية المستمرة والسجل الصحي للمستخدم.

### • الذكاء المالي المستقل:

مستشارون ماليون آليون يجمعون بين قواعد الاستثمار الصريحة وتحليل السوق بالتعلم العميق لتقديم نصائح استثمارية متكيفة.

### • الروبوتات ذاتية التعلم:

أنظمة تجمع بين القواعد المنطقية الواضحة للسلامة والتحكم مع التعلم التعزيزي للتكيف مع البيئات الجديدة والمتغيرة.

### • أنظمة دعم القرار الذكية:

أدوات مساعدة للقرارات الاستراتيجية المعقدة في الشركات والحكومات، تدمج المعرفة من خبراء متعددي التخصصات مع تحليل البيانات الضخمة.

## الرؤية المستقبلية

نظم تتعلم، تتكيف، وتشرح عملية استنتاجها بشفافية كاملة

النظم الخبيرة: من المفهوم إلى التطبيق

## التكامل مع تقنيات الذكاء الاصطناعي

### النظم الخبيرة + التعلم العميق

دمج قواعد المعرفة الصريحة مع قدرات التعلم العميق للشبكات العصبية، مما يسمح بإنشاء أنظمة تجمع بين المنطق المبني على القواعد والتعلم من البيانات.

### استخلاص المعرفة آلياً

استخدام تقنيات معالجة اللغة الطبيعية لاستخراج القواعد تلقائياً من الوثائق والمحادثات مع الخبراء، مما يقلل من عبء هندسة المعرفة.

### الاستدلال الاحتمالي

تطوير محركات استدلال تتعامل مع عدم اليقين باستخدام النظرية البيزية والمنطق الضبابي، مما يحاكي قدرة الخبير البشري على التعامل مع المعلومات غير الكاملة.



# خاتمة العرض التقديمي وشكر

## شكر وتقدير

أتقدم بجزيل الشكر والامتنان إلى كل من ساهم في إنجاح هذا البحث والعرض التقديمي:

### • أساتذتي الكرام

على توجيهاتهم القيمة ودعمهم المستمر خلال مراحل البحث والتنفيذ.

### • زملائي في الدراسة

على المناقشات المثمرة والأفكار الإبداعية التي أثرت هذا العمل.

### • المصادر والمراجع

التي استندت إليها في جمع وتحليل المعلومات عن النظم الخبيرة وتطبيقاتها.

## للتواصل والاستفسار

البريد الإلكتروني: [example@university.edu](mailto:example@university.edu)  
حساب لينكد إن: [linkedin.com/in/username](https://www.linkedin.com/in/username)

## الدروس المستفادة

### الفصل بين المعرفة والمنطق

المبدأ الأساسي في النظم الخبيرة هو الفصل بين المعرفة المتخصصة (قاعدة المعرفة) وآلية التفكير (محرك الاستدلال)، مما يتيح التطوير والصيانة بشكل أكثر فعالية.

### أهمية الشفافية والتبرير

النظم الخبيرة تتميز بقدرتها على شرح منطق قراراتها، مما يبني الثقة بينها وبين المستخدمين على عكس صناديق الذكاء الاصطناعي السوداء.

### البرمجة كائنية التوجه

استخدام OOP في تصميم النظم الخبيرة يسهل تمثيل المفاهيم المجردة كالحقائق والقواعد، ويتيح بناء أنظمة قابلة للتوسيع والتخصيص.