



以AI强化的RAD 开发人员

Sponsored by



Embarcadero is part of

Idera, Inc.

IDE & Tooling division:

- Embarcadero: Delphi, C++Builder, RAD Studio, InterBase
- UltraEdit: UltraEdit, UltraCompare, UltraFTP
- Whole Tomato: Visual Assist



whole tomato
Visual Assist

简介

Smart CodeInsight





RAD



What's New in <RAD Studio 12.2>

Smart CodeInsight：人工智能驱动的编码

使用 **AI LLM** 在 RAD Studio IDE 中进行编码

- 人工智能插件的**开放式架构**

为**三种在线解决方案**提供即用型支持

- OpenAI
- Gemini by Google
- Claude by Anthropic

和一种**离线解决方案**

- Ollama

即用型 **docker** 映像，可在 WSL2 中使用





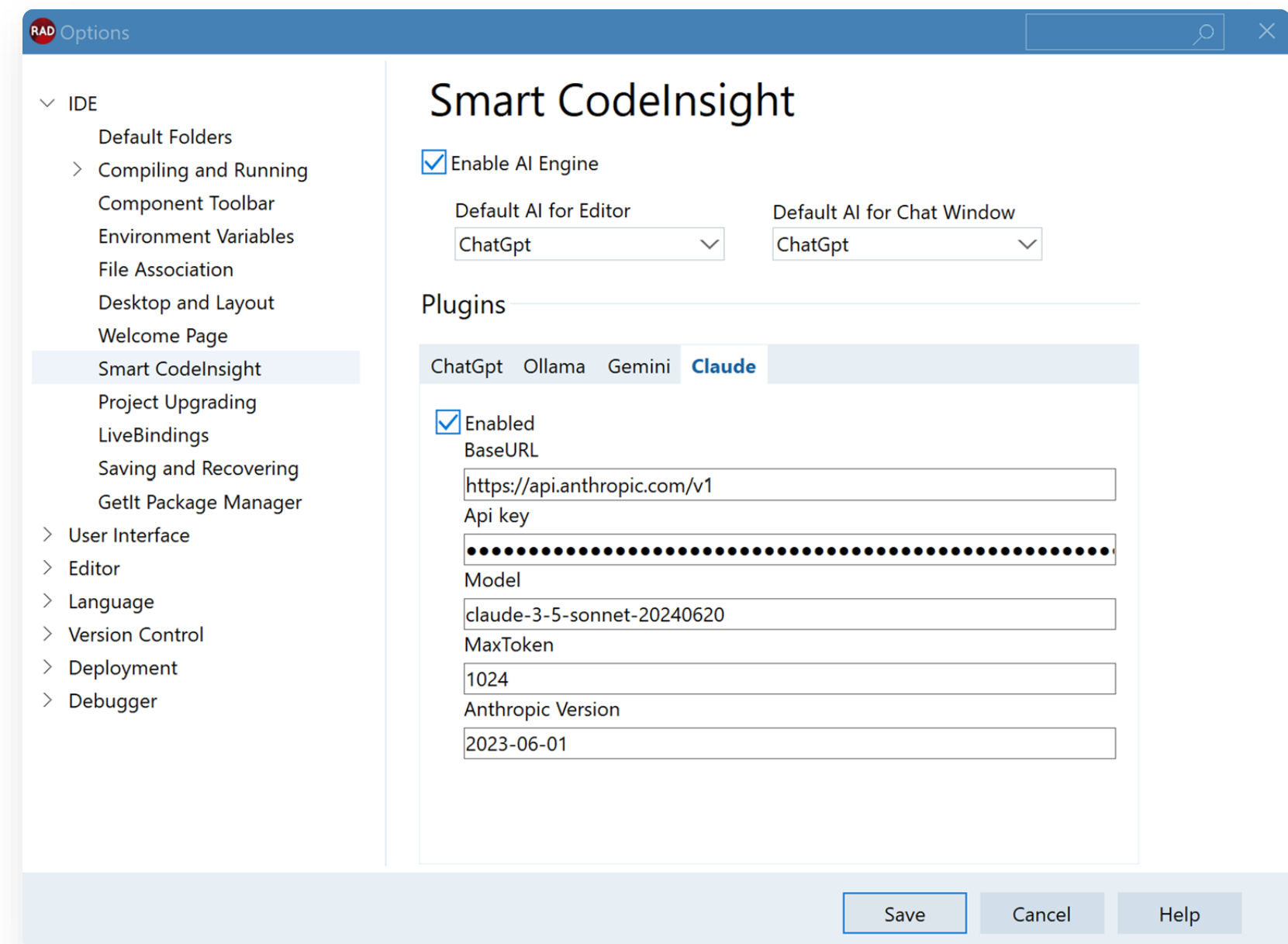
RAD



What's New in <RAD Studio 12.2>

Smart CodeInsight：人工智能驱动的编码

- 一切都是可选的并且预设为禁用
- 多个AI提供者，因此您可以选择要启用的一个或多个
- 本地安装的离线解决方案，最大程度保护隐私
- 你决定是否以及如何使用人工智能
...你甚至可以为你喜欢的解决方案创建额外的插件





RAD



What's New in <RAD Studio 12.2>

Smart CodeInsight：人工智能驱动的编码

● Smart CodeInsight

- 配置对话框中的模型列表
 - 始终更新可用模型列表
 - 直接从引擎刷新更新
- 新的Find Unit / Header (AI) 命令
 - 选择类型或符号并透过 AI 寻找所需uses或include的档案声明

Smart CodeInsight 选项

Options

Smart CodeInsight

☒ Enable Smart CodeInsight

Default AI for Editor:

Default AI for Chat Window:

Plugins

ChatGPT Ollama Gemini Claude

☐ Enabled

BaseURL:

Api key:

MaxToken: Temperature:

Model:

gpt-4o-mini
o1-preview
whisper-1
dall-e-3
babbage-002
omni-moderation-latest
omni-moderation-2024-09-26
tts-1-hd-1106

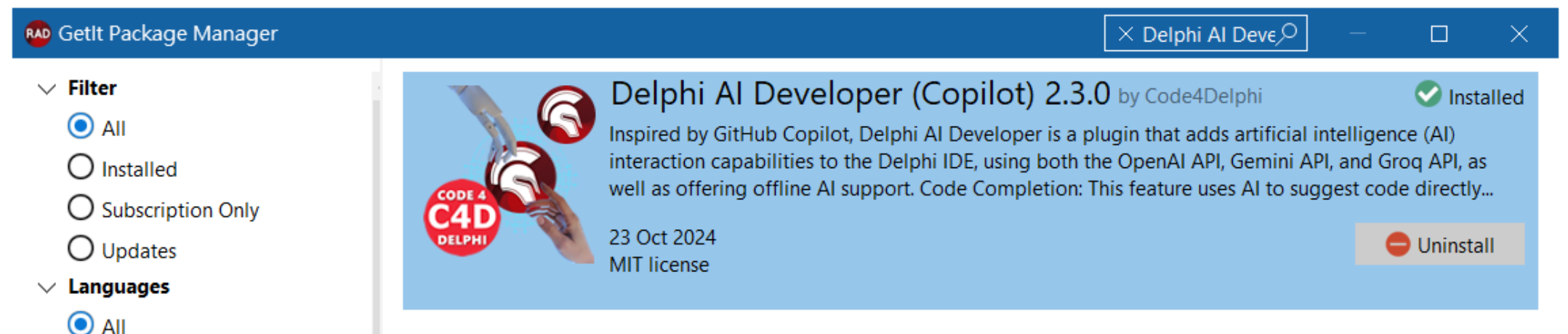
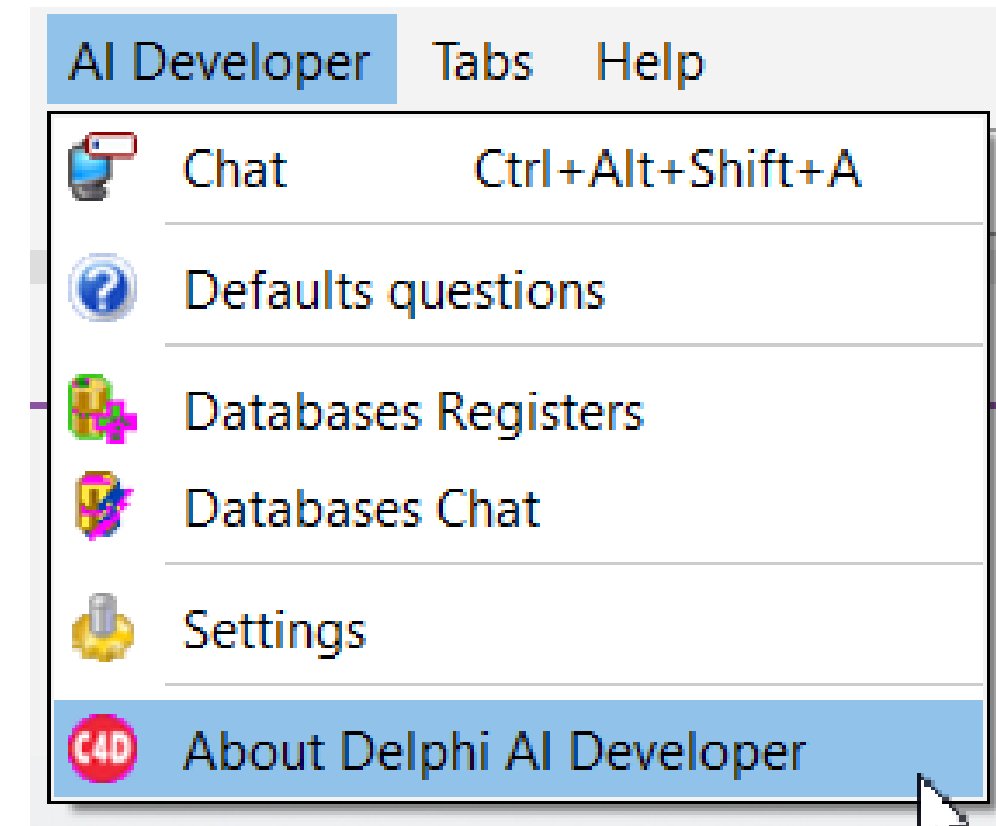
Save Cancel Help



其他RAD Studio的AI工具 **Delphi AI Developer (Copilot)**

Delphi AI Developer

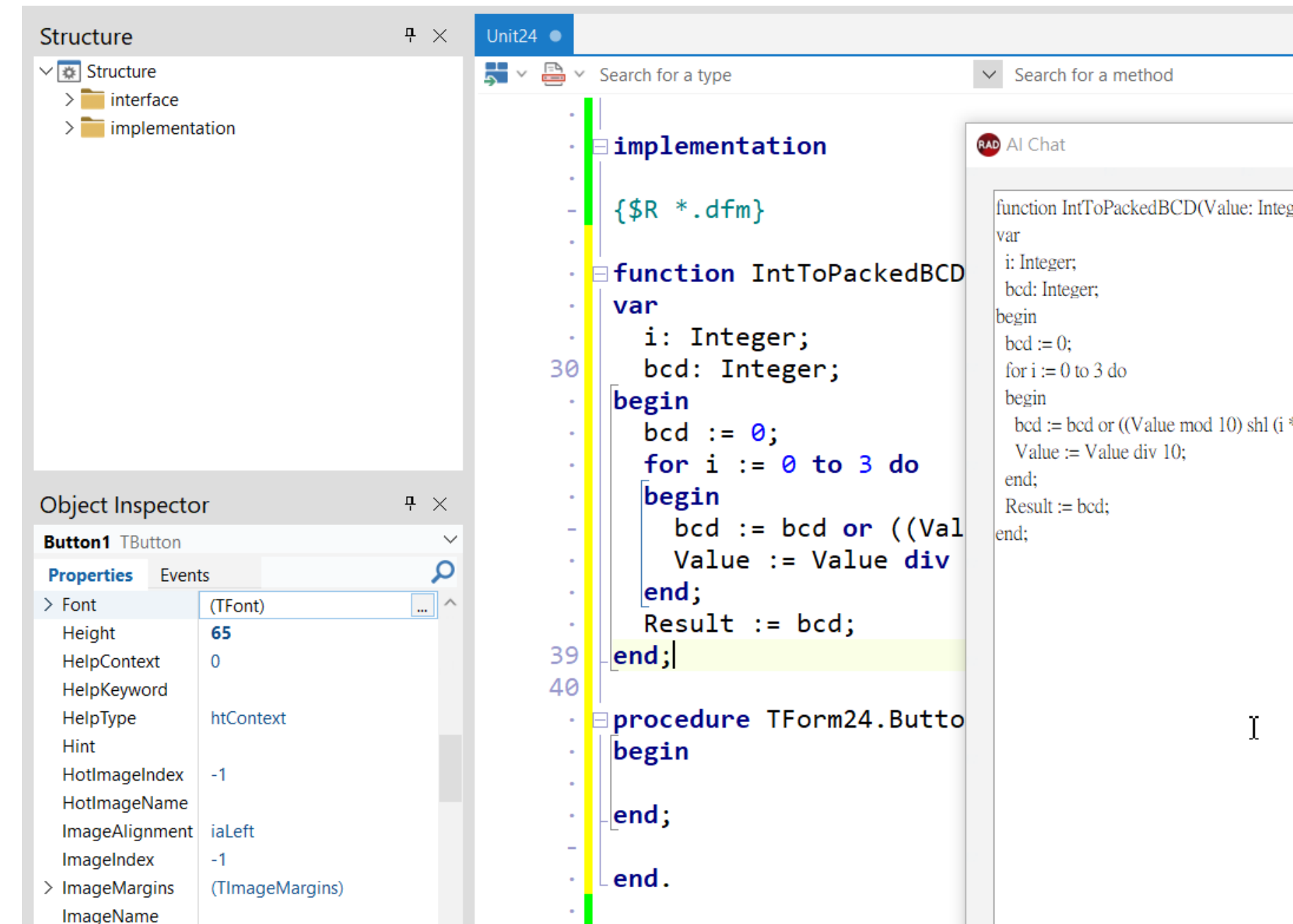
- Delphi AI Developer 是一个受 GitHub Copilot 启发的插件，它为 Delphi IDE 添加了人工智能 (AI) 互动功能，同时使用了 OpenAI API、Gemini API 和 Groq API，并提供脱机 AI 支援。
- 借助 Delphi AI Developer，您将获得生成和重构程序代码的协助，从而促进和加速开发。
- 直接在 IDE 中接收创建和改进程序代码的建议，并利用创建预定义问题的可能性来加快搜寻速度



Delphi AI Developer

- 提供Code Completion功能
- 提供AI整合功能, 例如在AI Chat中提问:

\$ 我是一名Delphi程序员, 请提供一个整数转换到packed BCD 的Delphi函数, 这个Delphi函数接受一个整数参数并且回传packed BCD形式的整数





RAD



What's New in <RAD Studio 12.2>

Delphi AI Developer

```
$ TVOCustomer = class  
private  
    FCustomerName: string;  
    FPhone: string;  
    FAddress: string;  
    FEmail: string;  
public  
    property CustomerName: string read FCustomerName write FCustomerName;  
    property Phone: string read FPhone write FPhone;  
    property Address: string read FAddress write FAddress;  
    property Email: string read FEmail write FEmail;  
end;
```



使用**AI**成为更好的**RAD** 开发人员



RAD



What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员

- *AI 可以成为一个极佳的Pair Programmer*
- *AI 可以帮助我们进行开发工作*
- *AI 可以帮助我们结合设计样例, 程序代码, 测试和部署的工作*
- *AI 可以帮助我们结合DevOps/ DevSecOps*



RAD



What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员

- 从人工智能到Transformer

Artificial intelligence

Any technique that allows a computer to mimic human behavior

Machine learning

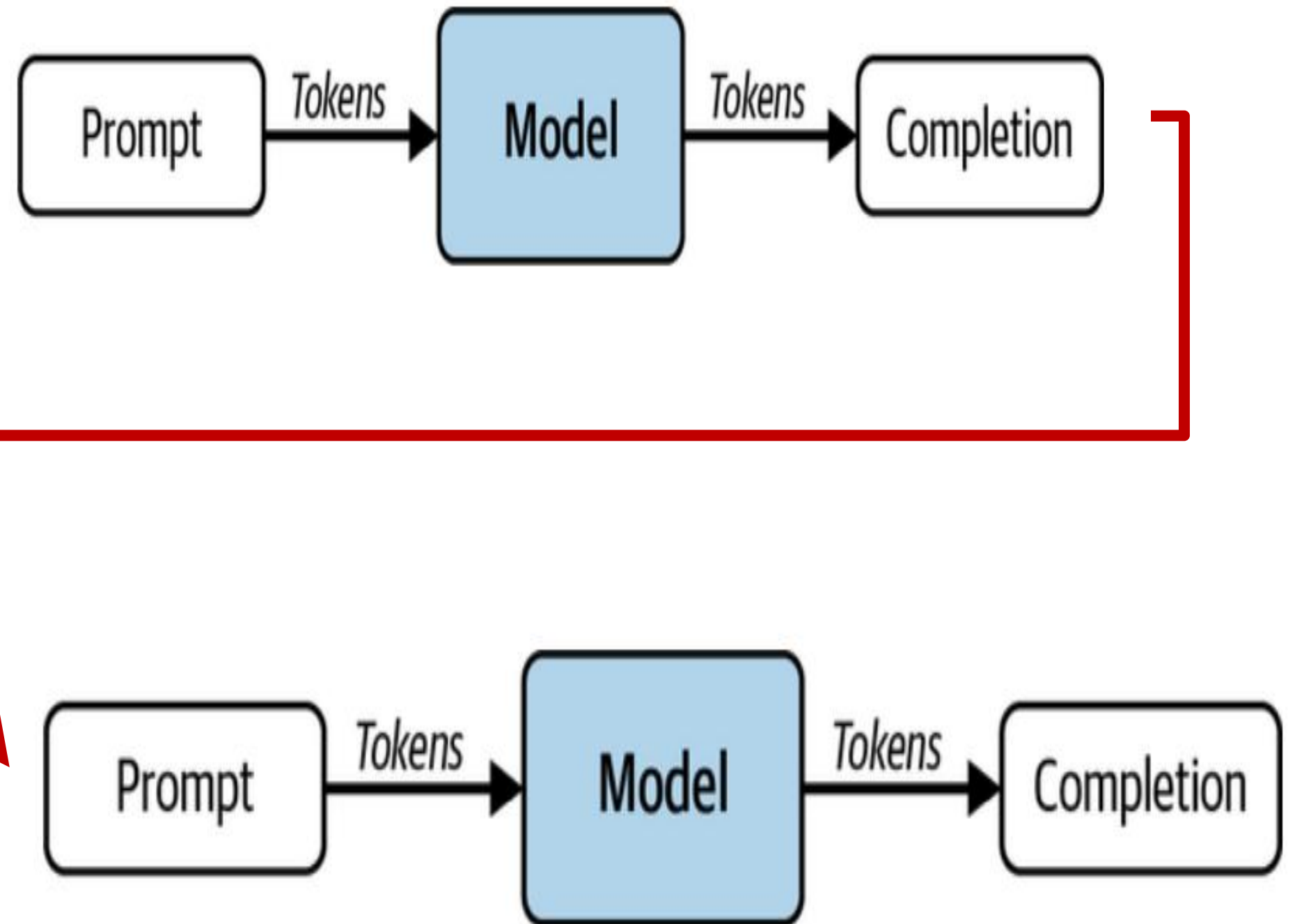
Ability to learn without explicit programming

Deep learning

Ability to extract patterns from data using artificial neural networks

Transformers

GPT models are based on this





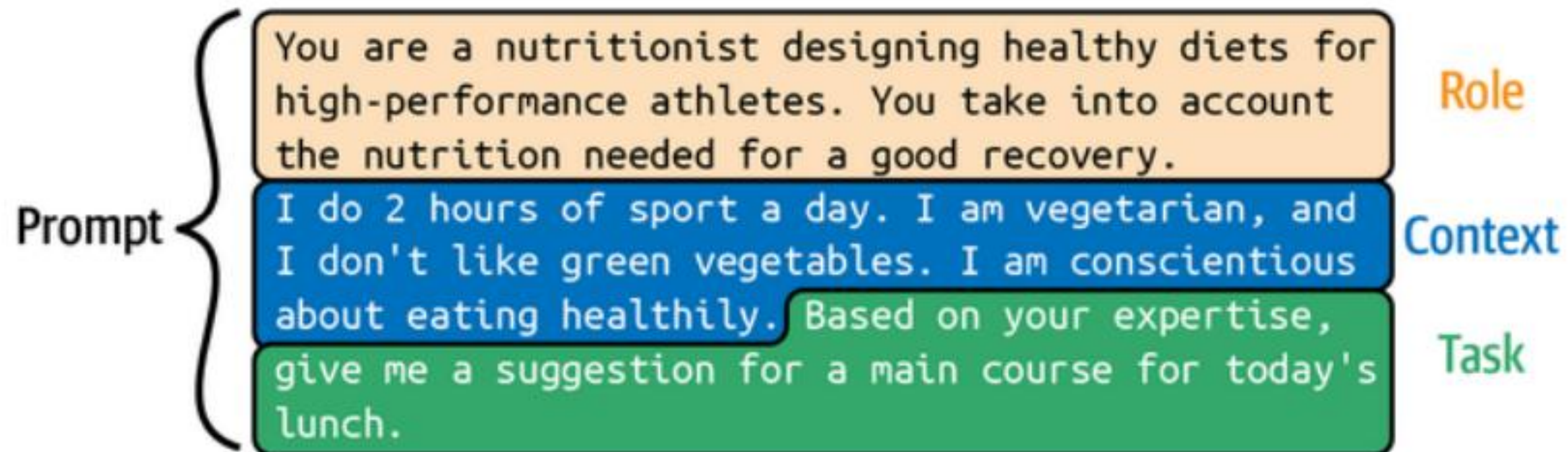
RAD



What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员

- **Prompt Engineering** - 是一门专注于开发最佳实践的学科为LLM建立最佳输入，并尽可能以程序设计方式产生理想的输出
- 设计包含角色、上下文和内容的有效提示任务





使用AI成为更好的RAD 开发人员

◦ *提示公式是提示的具体格式，一般为由3个主要元素组成*

➤ *任务：清晰简洁地说明提示内容要求模型生成输出内容。*

◦ ➤ 说明/上下文：应遵循的说明生成文字时的模型。

➤ 角色：模型在生成时应扮演的角色文字。

◦



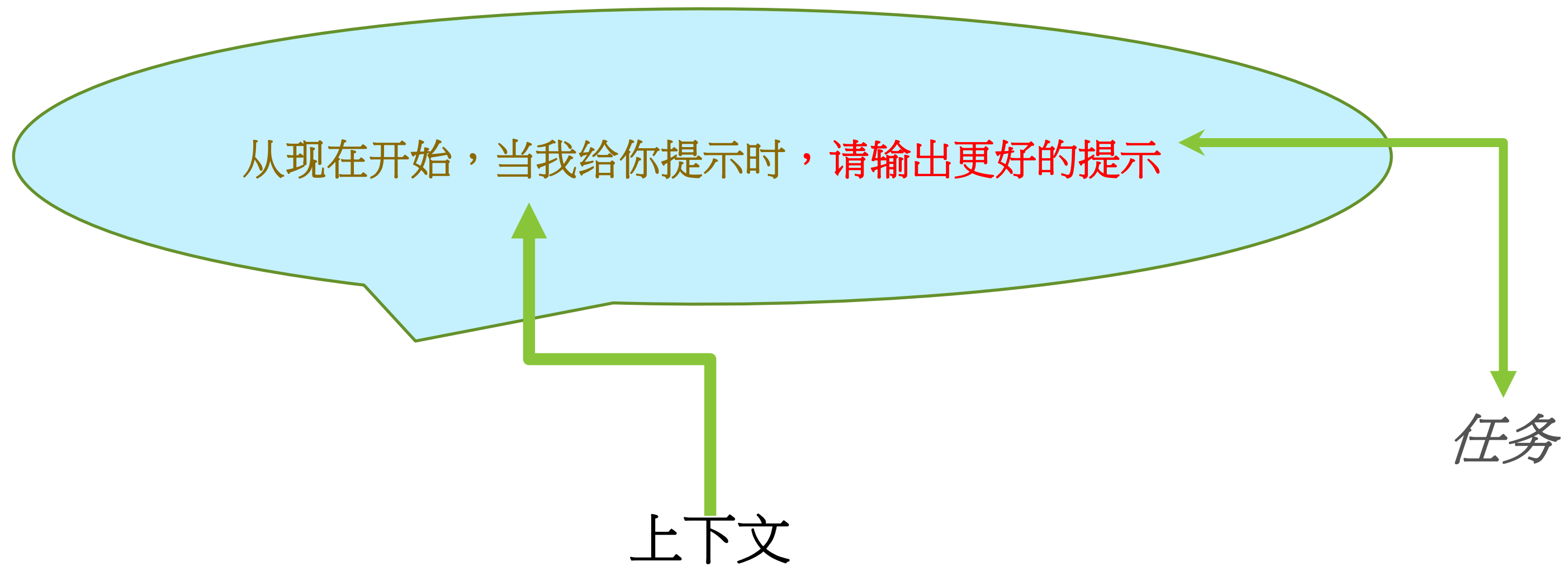
RAD



What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员

提示公式





RAD

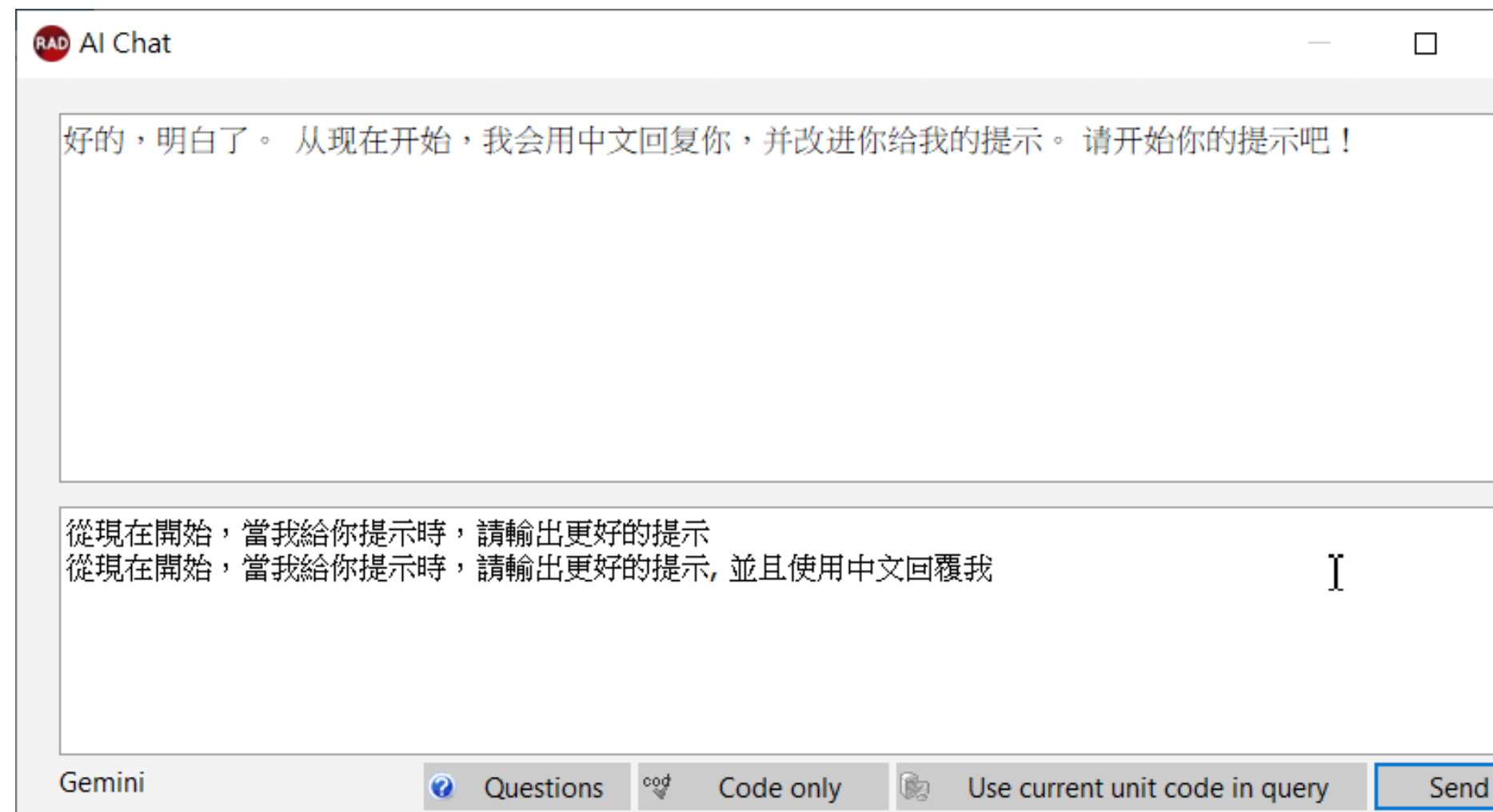


What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员

- 从现在开始，当我给你提示时，请输出更好的提示

- 从现在开始，当我给你提示时，请输出更好的提示





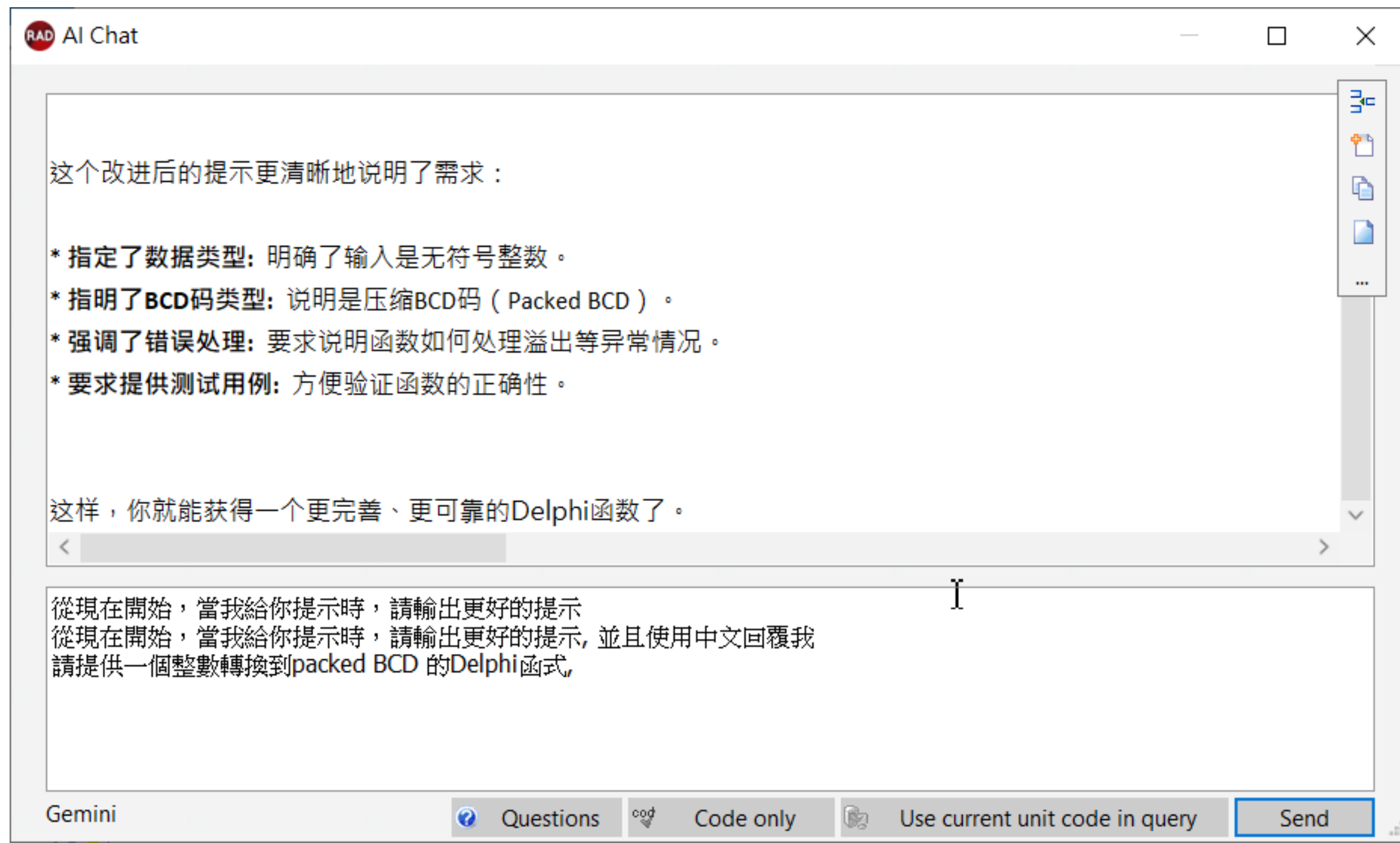
RAD



What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员

- 请提供一个整数转换到packed BCD 的Delphi函数,





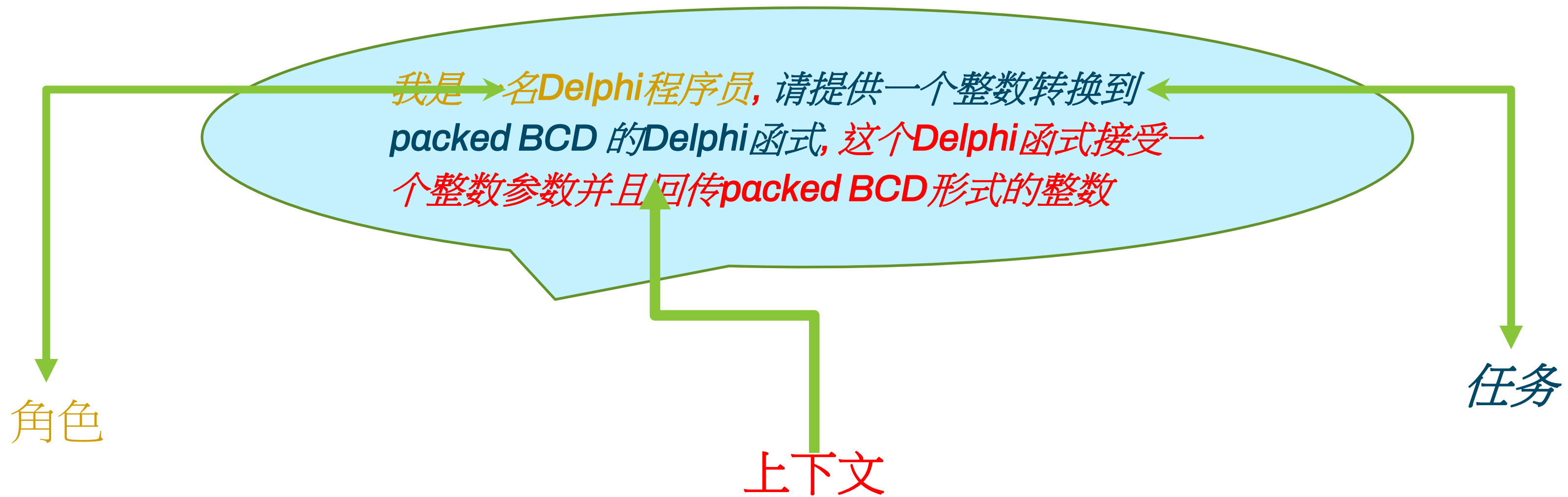
RAD



What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员

提示公式





RAD



What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员

- 我是一名Delphi程序员, 请提供一个整数转换到packed BCD 的Delphi函数, 这个Delphi函数接受一个整数参数并且回传packed BCD形式的整数

RAD AI Chat

```
function IntToPackedBCD(Value: Integer): Integer;  
var  
  i: Integer;  
  bcd: Integer;  
begin  
  bcd := 0;  
  for i := 0 to 3 do  
  begin  
    bcd := bcd or ((Value mod 10) shl (i * 4));  
    Value := Value div 10;  
  end;  
  Result := bcd;  
end;
```

從現在開始，當我給你提示時，請輸出更好的提示

從現在開始，當我給你提示時，請輸出更好的提示, 並且使用中文回覆我

請提供一個整數轉換到packed BCD 的Delphi函式,

我是一名Delphi程式師, 請提供一個整數轉換到packed BCD 的Delphi函式, 這個Delphi函式接受一個整數參數並且回傳packed BCD形式的整數

使用AI成为更好的RAD 开发人员

- 现在就让我们使用一个简单的范例来看看如何使用AI来帮助我们开发应用程序
- 这个简单的范例将结合设计样例
- 让我们快速开发一个“个人股票管理系统”
- 在过程中我们还可以比较**Gemini, ChatGPT, DeepSeek**的回答
- 我们还可以使用**RAG/Fine-Tuning**继续整合后续测试, 其他平台(例如**Web**)的开发工作, 但这是一个Webinar, 我们就不花太多的时间讨论后续的开发工作
- 注意, “由于**AI Provider**的免费使用有限制, 因此我们在一定提问之后就可能无法使用, 请各位见谅!”
- 现在就开始吧!



使用AI成为更好的RAD 开发人员

○ 个人股票管理系统

- 1) 从现在开始，当我给你提示时，请输出更好的提示，并且使用中文回复我
- 2) 假设您是软件架构师，并且您正在设计一个个人存股管理系统，使用Delphi. 此系统将能够管理所有存股信息，包括股票名称，股票代码，购入日期，购入单价，购入股数，手续费，购入券商. 还需要记录股票配息信息，包括股票代码，股票配息日期，配息金额，配息殖利率. 此外还要记录股票售出信息，包括股票名称，股票代码，售出日期，售出金额，售出股数，手续费. 最后系统还要能提供每种股票的总张数，平均成本信息. 它应该使用六边形架构. 记得加强我的提示
- 3) 显示此应用系统的Delphi类别
- 4) 此系统将使用Interbase做为数据库，请根据应用系统的Delphi类别产生Interbase数据库架构
- 5) 请产生此系统接口与Adapter类别
- 6) TInterbaseStockRepository 类别将使用FireDAC存取Stocks数据表，请显示TInterbaseStockRepository 实作程序代码

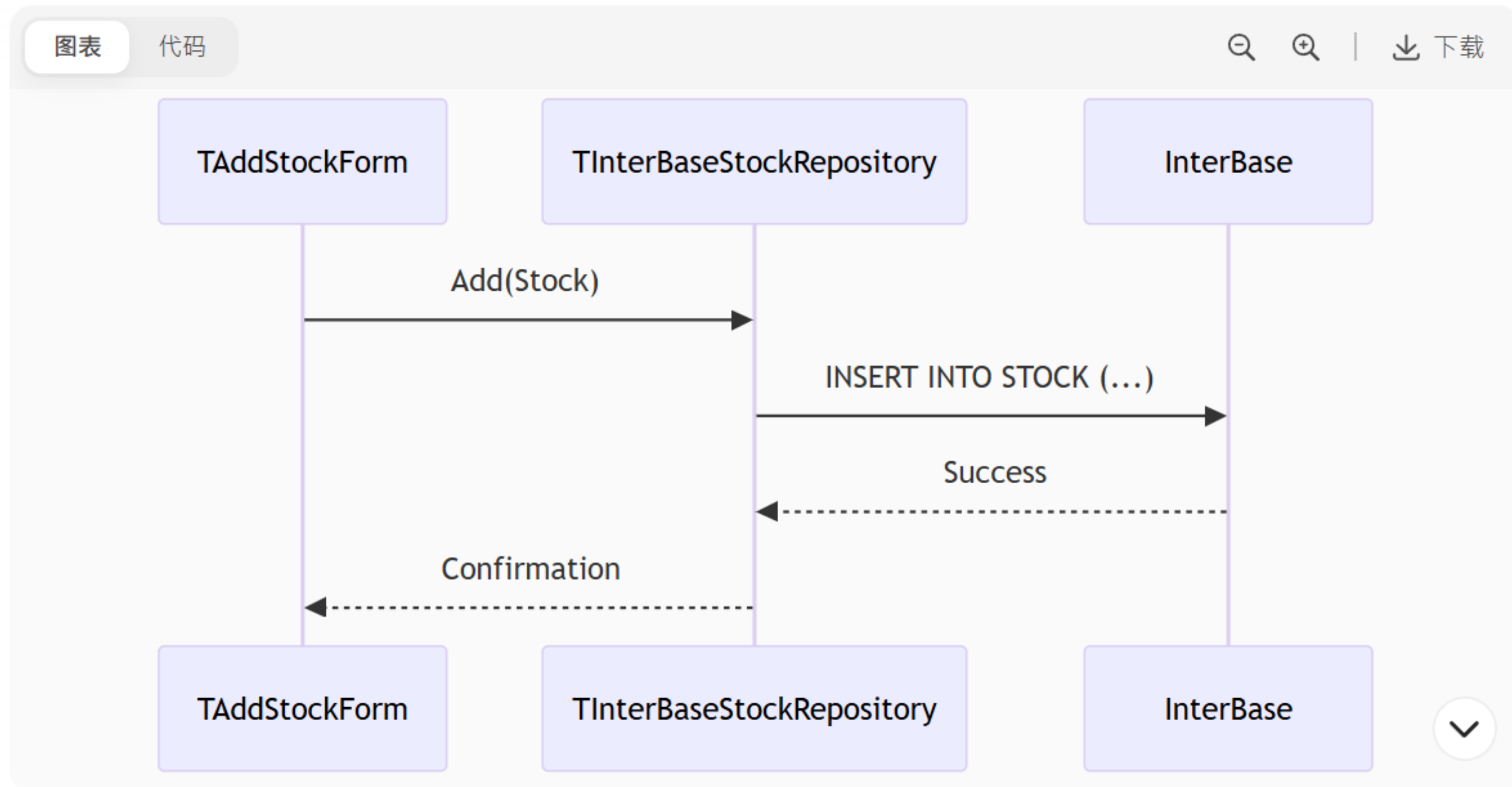


RAD



What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员





RAD



What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员

○ 个人股票管理系统

- 7) 请显示TDatabaseStockRepository 的实作程序代码, 使用FireDAC和刚才的InterBase数据库
- 8) 请产出前述所有数据表20笔范例数据的SQL, 并且维持例数据之间的关联性
- 9) 我应该使用什么测试架框来测试上述的专案
- 10) 请给我看这个专案的Mermaid 类别图
11. 我想开发此系统的Web界面, 请建议应该使用什么解决方案, 使用Delphi
12. 如果我想使用TMS 的Web Core呢?
13. 请详细说明如何与 Delphi 后端整合

1. 股票基本資料 (STOCK)

sql

复制 下载

```
-- 台灣上市公司股票範例
```

```
INSERT INTO STOCK (STOCK_ID, STOCK_CODE, STOCK_NAME) VALUES (1, '2330.TW', '台積電');
INSERT INTO STOCK (STOCK_ID, STOCK_CODE, STOCK_NAME) VALUES (2, '2317.TW', '鴻海');
INSERT INTO STOCK (STOCK_ID, STOCK_CODE, STOCK_NAME) VALUES (3, '2454.TW', '聯發科');
INSERT INTO STOCK (STOCK_ID, STOCK_CODE, STOCK_NAME) VALUES (4, '2882.TW', '國泰金');
INSERT INTO STOCK (STOCK_ID, STOCK_CODE, STOCK_NAME) VALUES (5, '1303.TW', '南亞');
```

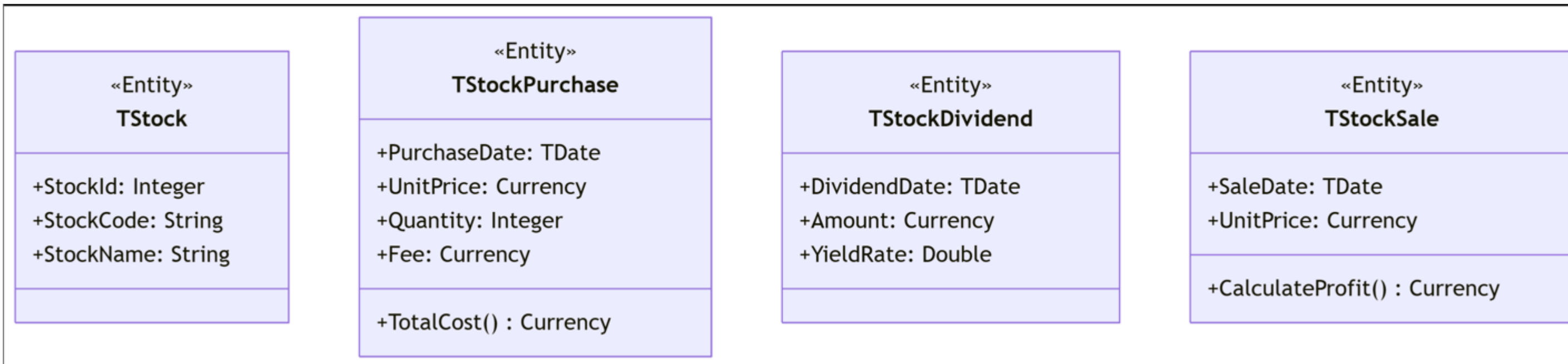


RAD



What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员



TStock "1" *-- "0..*" TStockPurchase : 包含

TStock "1" *-- "0..*" TStockDividend : 包含

TStock "1" *-- "0..*" TStockSale : 包含

IStockRepository <|.. TInterBaseStockRepository : 实现

IPurchaseRepository <|.. TInterBasePurchaseRepository : 实现

TMainForm --> IStockRepository : 依赖

TAddStockForm --> IStockRepository : 依赖

TInterBaseStockRepository --> TFDConnection : 使用

TInterBaseStockRepository --> TFDDataSet : 返回



RAD



What's New in <RAD Studio 12.2>

使用AI成为更好的RAD 开发人员

- *RAD Studio* 应该会持续强化AI的功能, 期待在未来的版本中看到更多AI的功能和*RAD Studio*的整合

○

Q&A?





以AI强化的RAD 开发人员
感谢您的参加!