



# 안영제 / 시골프로그래머

---

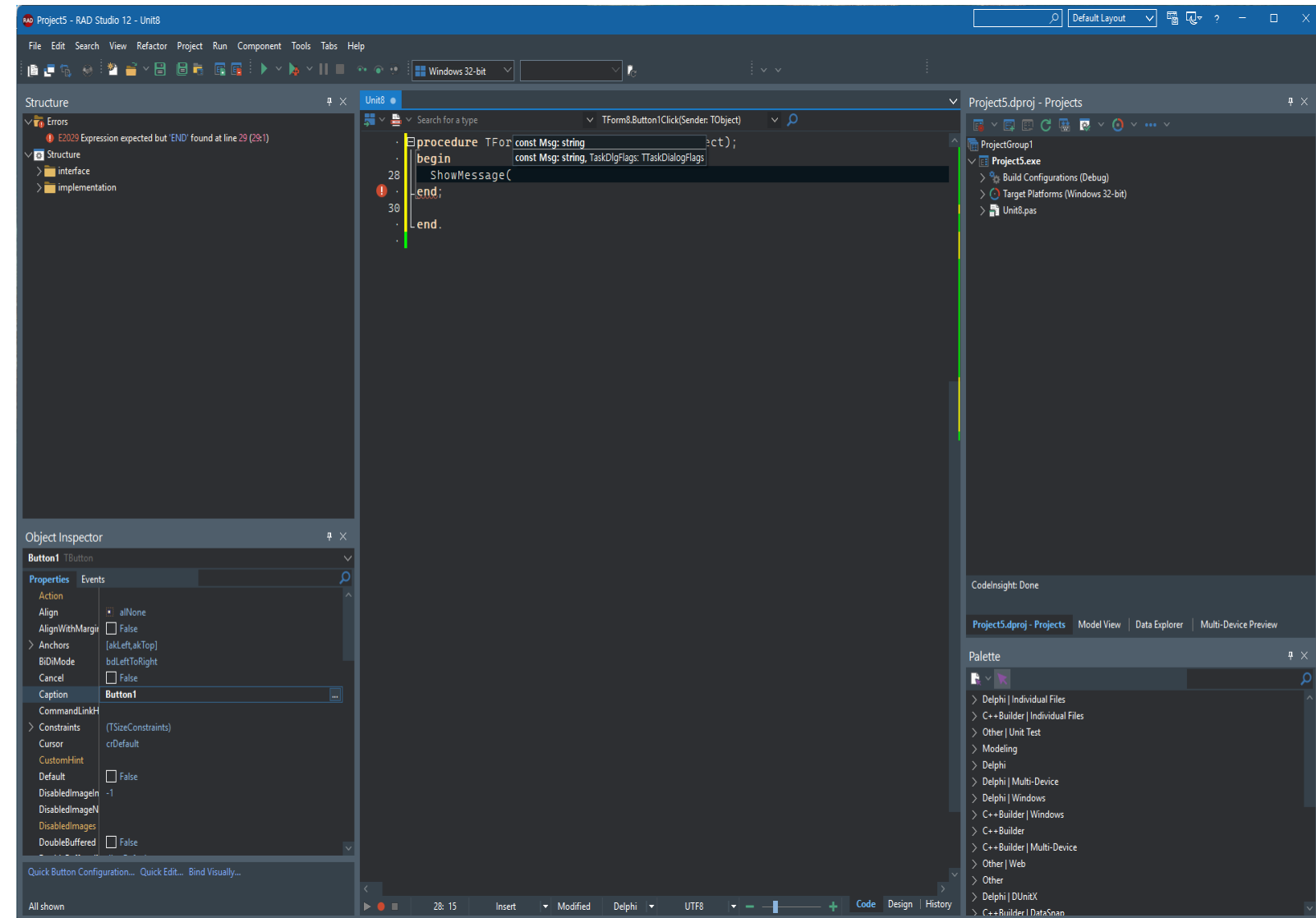
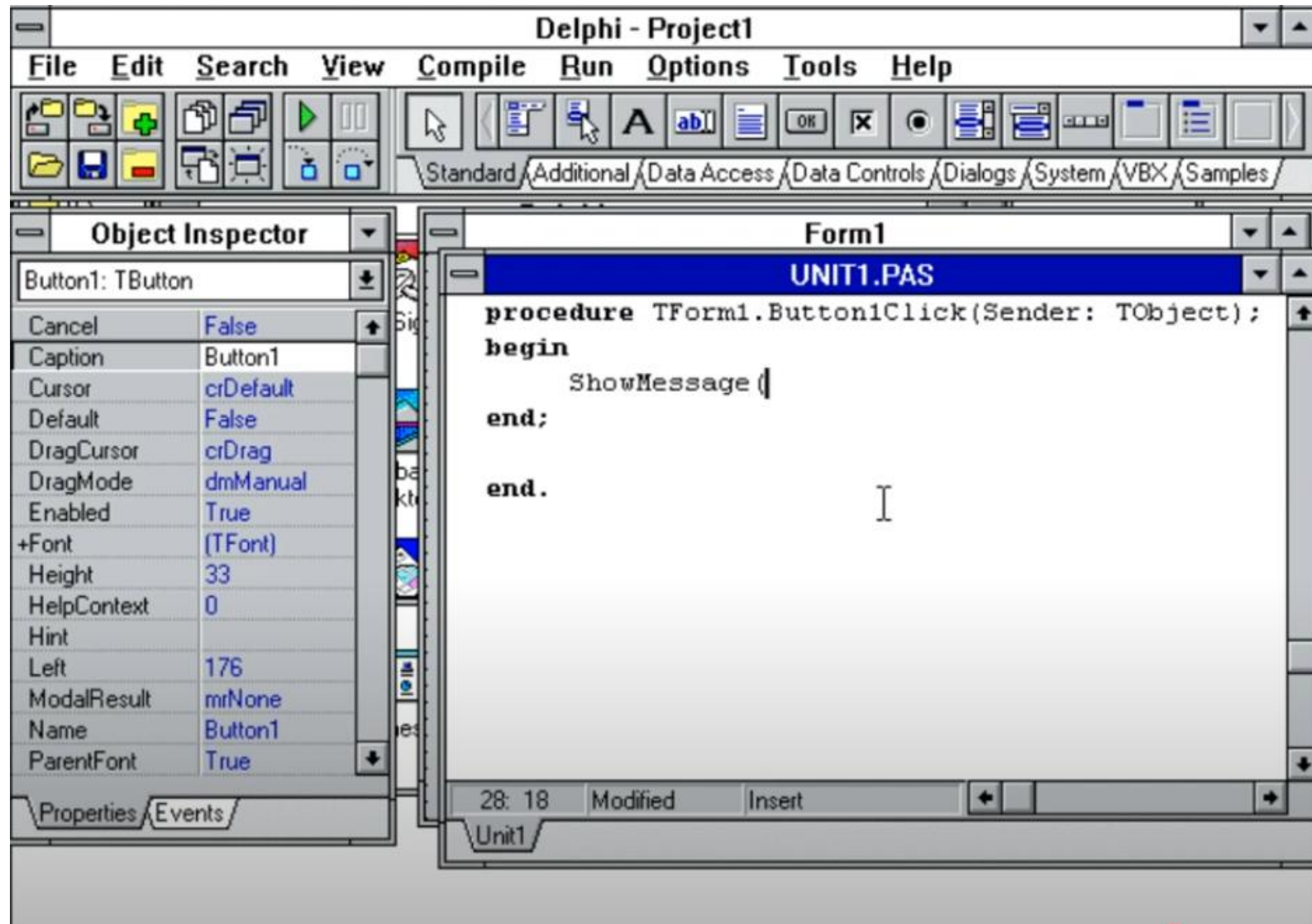
폴빛컴퓨팅 대표  
개발자  
컨설턴트  
델마당 운영자

<https://github.com/civilian7>

<https://github.com/delmadang>



# 30년, 눈 깜짝할 사이에...





# DELPHI 1.0



# 변한 것, 변하지 않은 것

## 변한 것

- 현대적 IDE와 언어 스펙
- 멀티 OS 지원

## 변하지 않은 것

- 간결함
- 강력함
- 여러분의 사랑



# Agenda

눈떠보니 AI 시대

AI 시대를 개발자는 어떻게 맞을 것인가?

IDE에 AI 어시스턴트 연결하기

- 자연어로 코드 생성해보기
- 기타 기능 훑어보기

샘플 프로그램

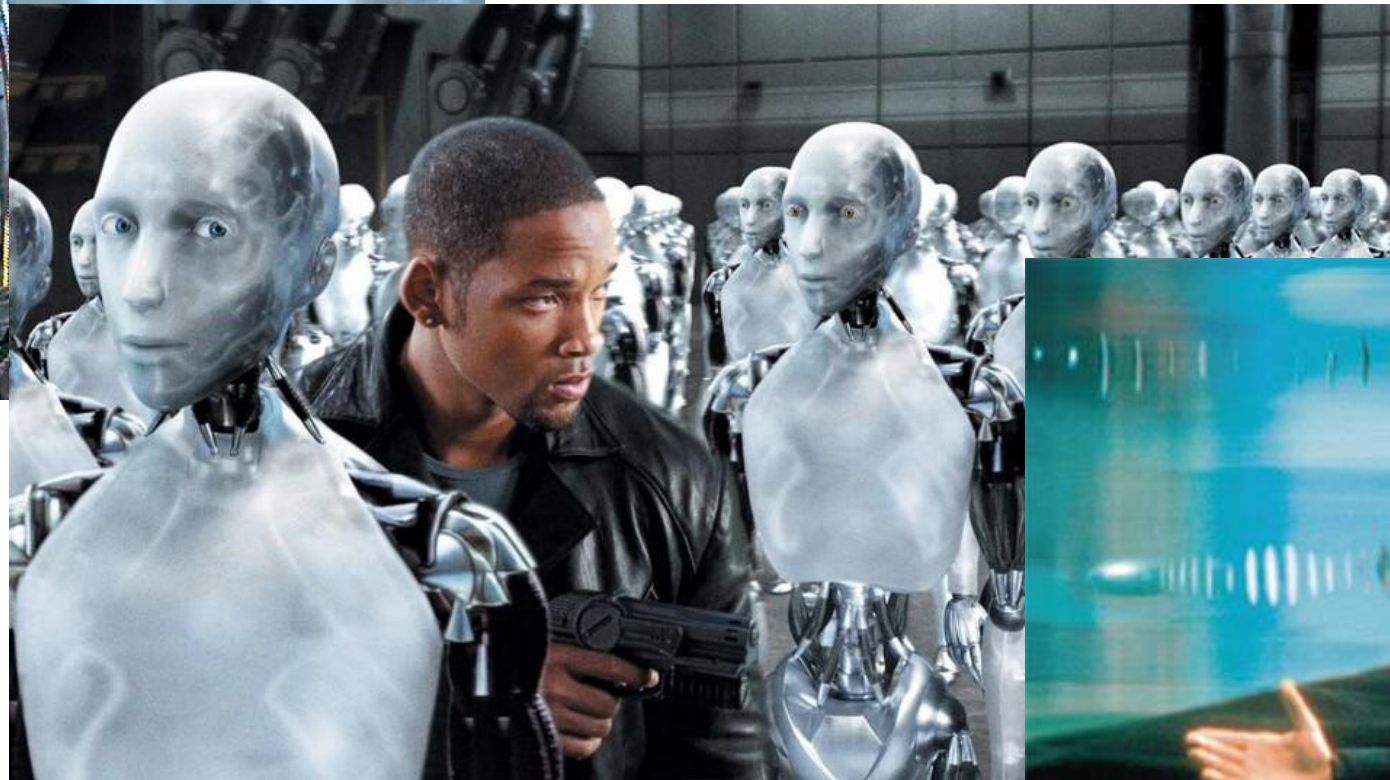
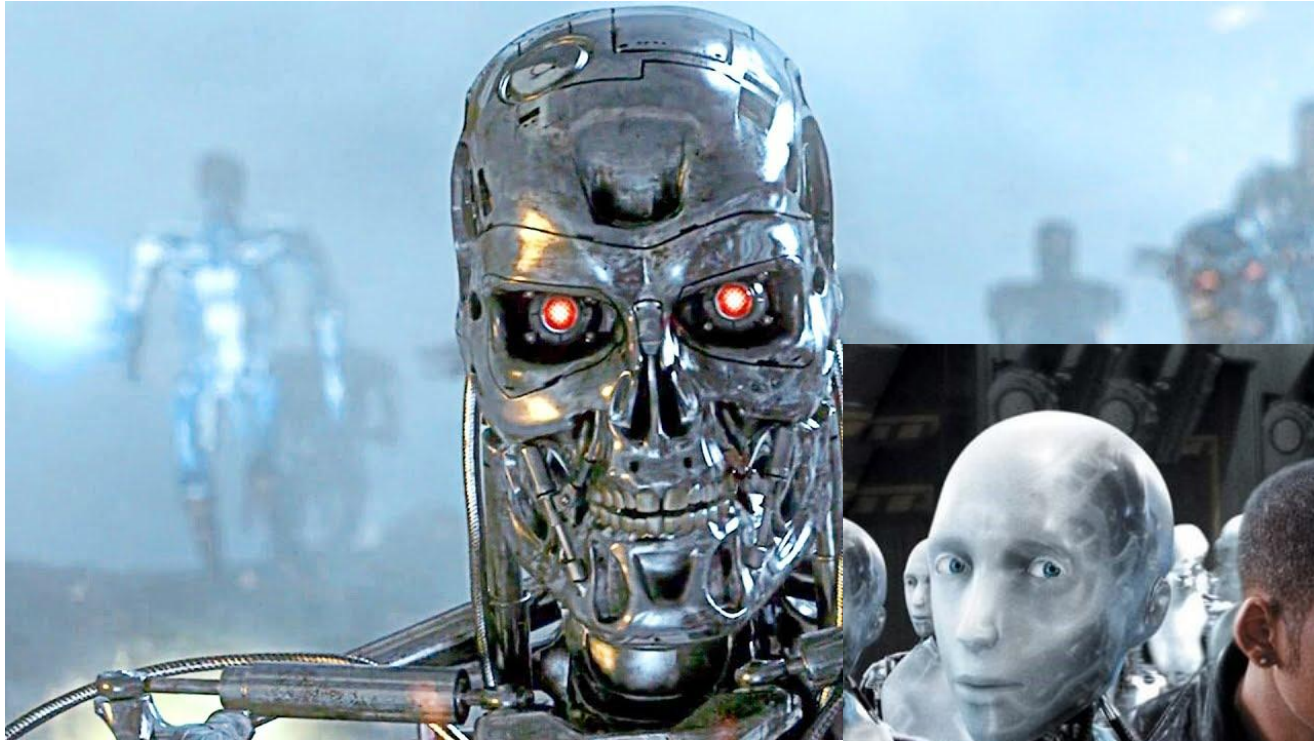
- 프로그램에 AI 어시스턴트 연결하기
- 자연어 질의 기능 만들어보기
- Claude를 사용해서 영수증에서 텍스트 뽑아오기

A heart-shaped logo with a blue background. Inside the heart, there's a white outline of a classical statue head (Delphi). The text "It's Delphi's" is in white, and "30th!" is in large white font with a red and white circular graphic around the "0".

It's Delphi's  
30th!



# AI, 인류를 말살할거야...





# AI, 대중화되다 // 최초이자 마지막 인간



YTN

NEWS



# AI, 이제 대세

한때 공상과학 속 이야기였던 인공지능(AI)은 이제 우리 일상의 중심이 되었다.

- 스마트폰 음성 비서 (시리, 빅스비, 헤이 구글, 알렉사 등)
- 자동 번역 (DeepL)
- 자율주행차 (테슬라)
- 챗봇
- 코딩 (Copilot, Cursor, Windsurf)

등, AI는 다양한 분야에서 활약하고 있다.



# AI, 우리는 어떻게 사용해야 할까?

## AI는 개발자의 적이 아니라 친구

- 개발자는 AI를 활용해 생산성을 높일 수 있다
- 단순하고 반복적인 작업을 줄여 창의적인 문제 해결에 집중할 수 있다
- AI 도구 활용 능력을 높이는 것이 미래 개발자의 핵심 역량이 될 것이다

## 코딩에서 문제 해결로

- 코딩은 AI가, 개발자는 아키텍처 설계 및 최적화에 집중

## AI와 협업

- AI 도구를 효과적으로 사용하는 방법에 익숙해져야 한다



# 코드 작성

## AI 코드 어시스턴트

- Github Copilot, Cursor, Windsurf : 자연어로 설명하면 코드를 생성
- Tabnine, Codeium : AI 기반 코드 자동 완성

## 자동 코드 변환 및 최적화

- 프로그래밍 언어간 변환 예) C/C++ -> PASCAL
- 코드 스타일 최적화 및 리팩토링

<https://codeium.com/windsurf>  
<https://www.cursor.com/>



# 디버깅 및 오류 해결

## AI 기반의 오류 분석 및 수정

- Snylc, DeepCode : 보안 취약점 자동 감지
- Amazon CodeWhisperer : 오류 코드 추천 및 수정

## 로그 및 성능 분석

- AI Ops : 서버 로그, 에러 로그 분석 및 자동 대응
- New Relic, Datadog : 성능 모니터링 및 이상 감지



# 테스트 자동화

## 테스트 코드 생성

- Diffblue Cover : AI 기반의 자동 단위 테스트 생성
- Testim, Mabl : 자동화 테스트 스크립트 생성 및 실행

## 버그 예측 및 회귀 테스트

- AI가 코드 변경을 분석하고 버그 가능성 예측
- 기존 기능에 영향을 주는 회귀 테스트 자동화

# 델파이와 AI 통합

## 주요 LLM을 IDE에서 지원

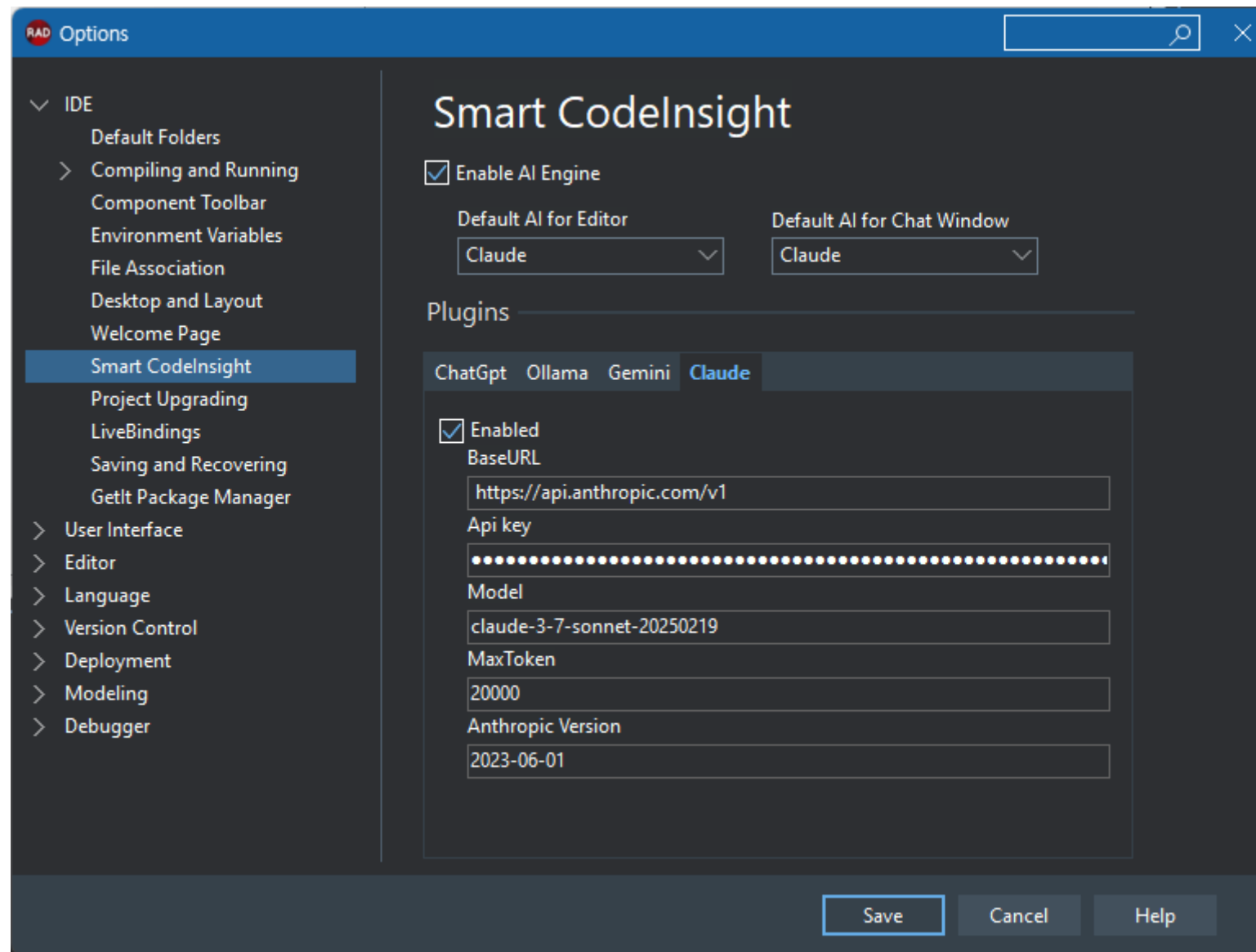
- 내장된 **Smart Inspector**를 통해 주요 LLM 사용가능 (APIKey 필요)
- 다양한 서드파티를 통해 LLM 확장 가능

## 델파이와 AI 의 통합은 제한적

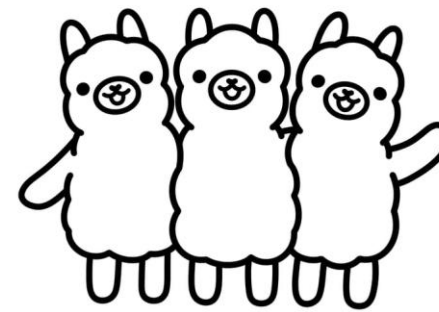
- AI 친화적인 언어는 파이썬, 자바스크립트 등 인터프리터 방식이 대세
- 학습한 코드의 절대치가 매우 부족 – 오픈소스 코드의 부족이 원인
- WYSWYG 방식의 UI 디자이너가 오히려 장벽
- 더디지만, 통합의 길로 나아갈 것.



# Smart Inspector 설정



Meta

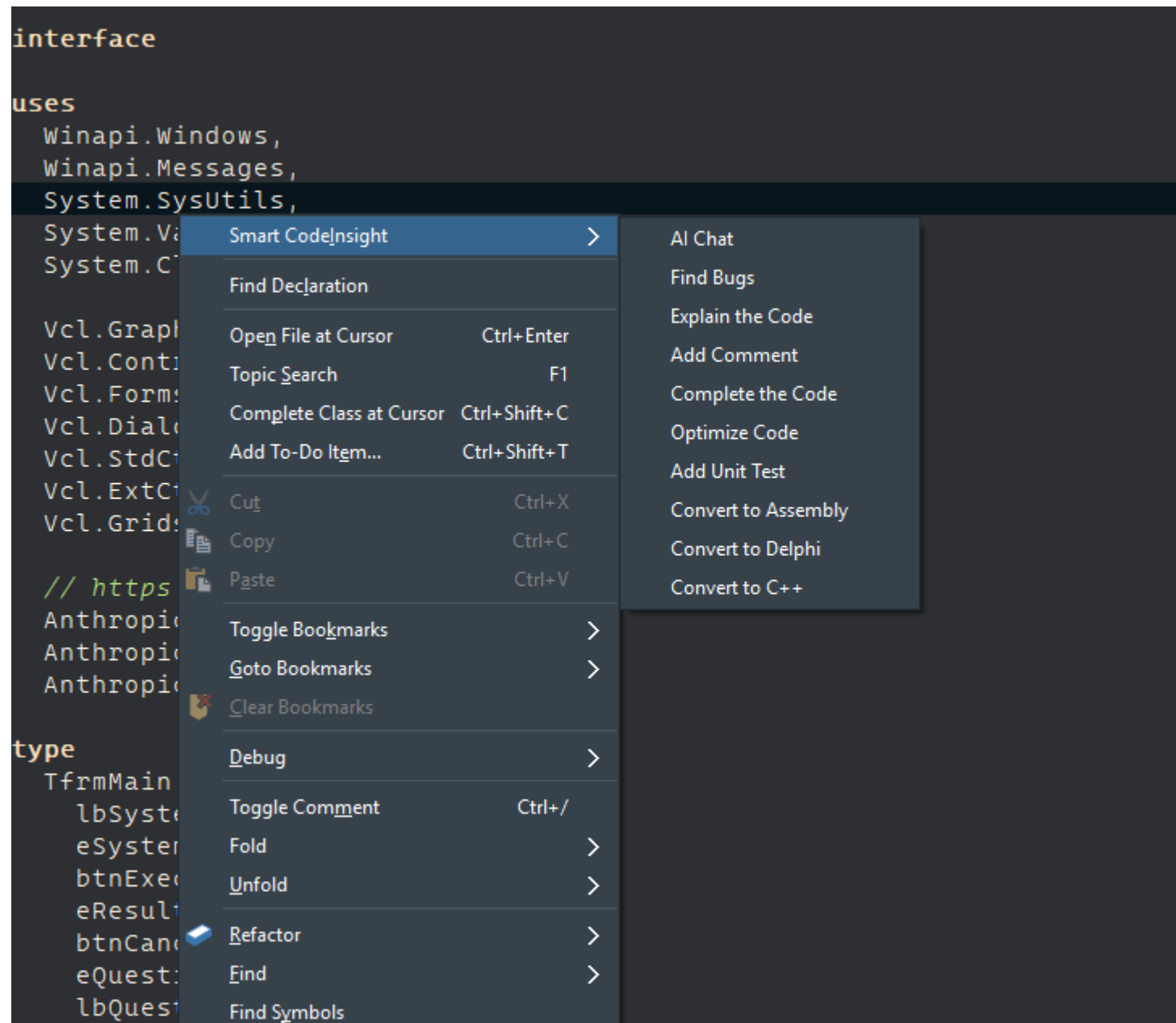


ollama run llama3



[https://docwiki.embarcadero.com/RADStudio/Athens/en/Smart\\_CodeInsight\\_Options](https://docwiki.embarcadero.com/RADStudio/Athens/en/Smart_CodeInsight_Options)

# Smart Inspector // from 12.2

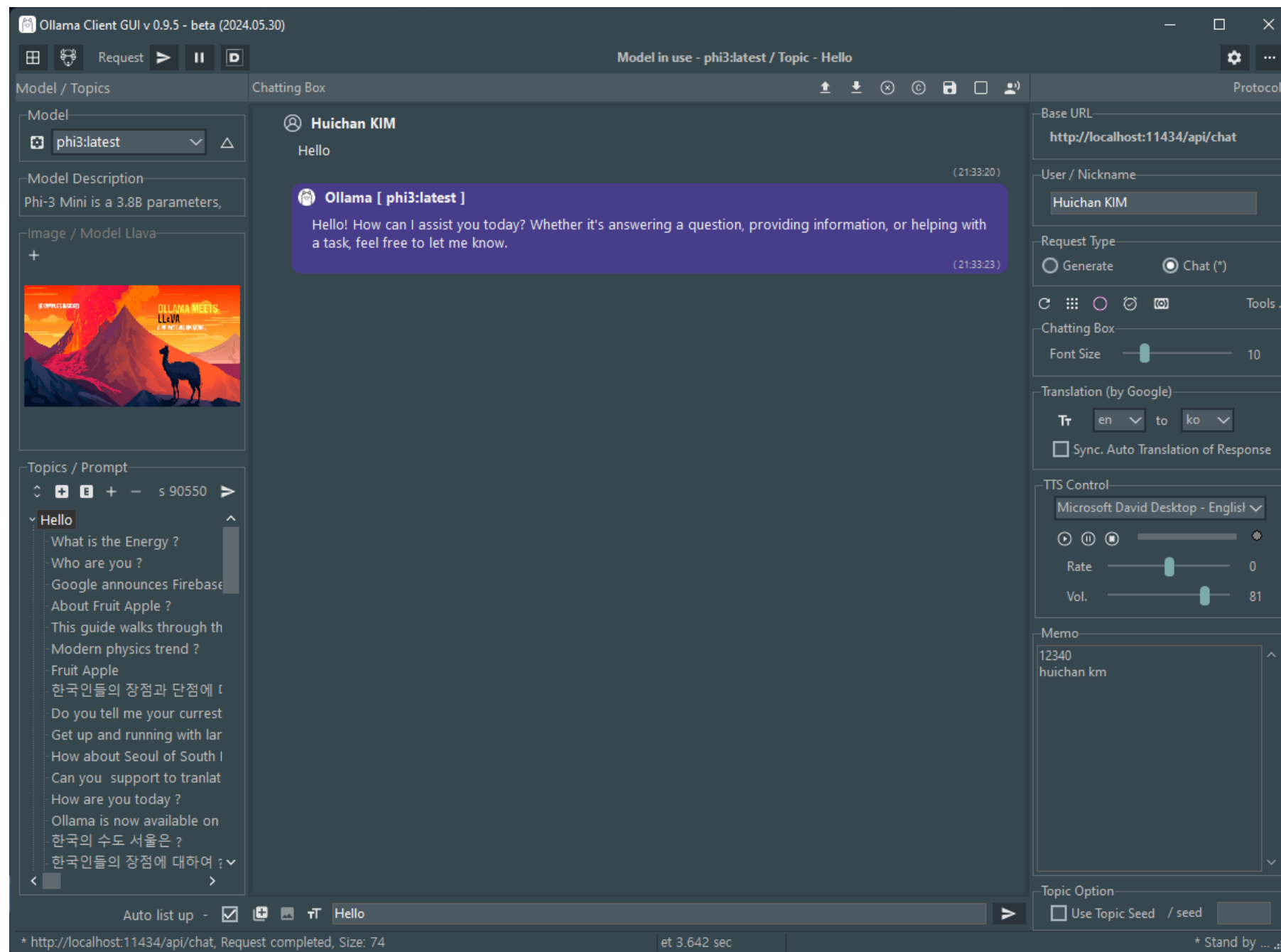


자연어로 질의할 수 있는 AI 챗  
소스코드 분석으로 버그 검토  
소스코드 해설  
주석 삽입  
코드 완성  
코드 최적화  
유닛 테스트 추가  
코드 변환 Assembly, Delphi, C++





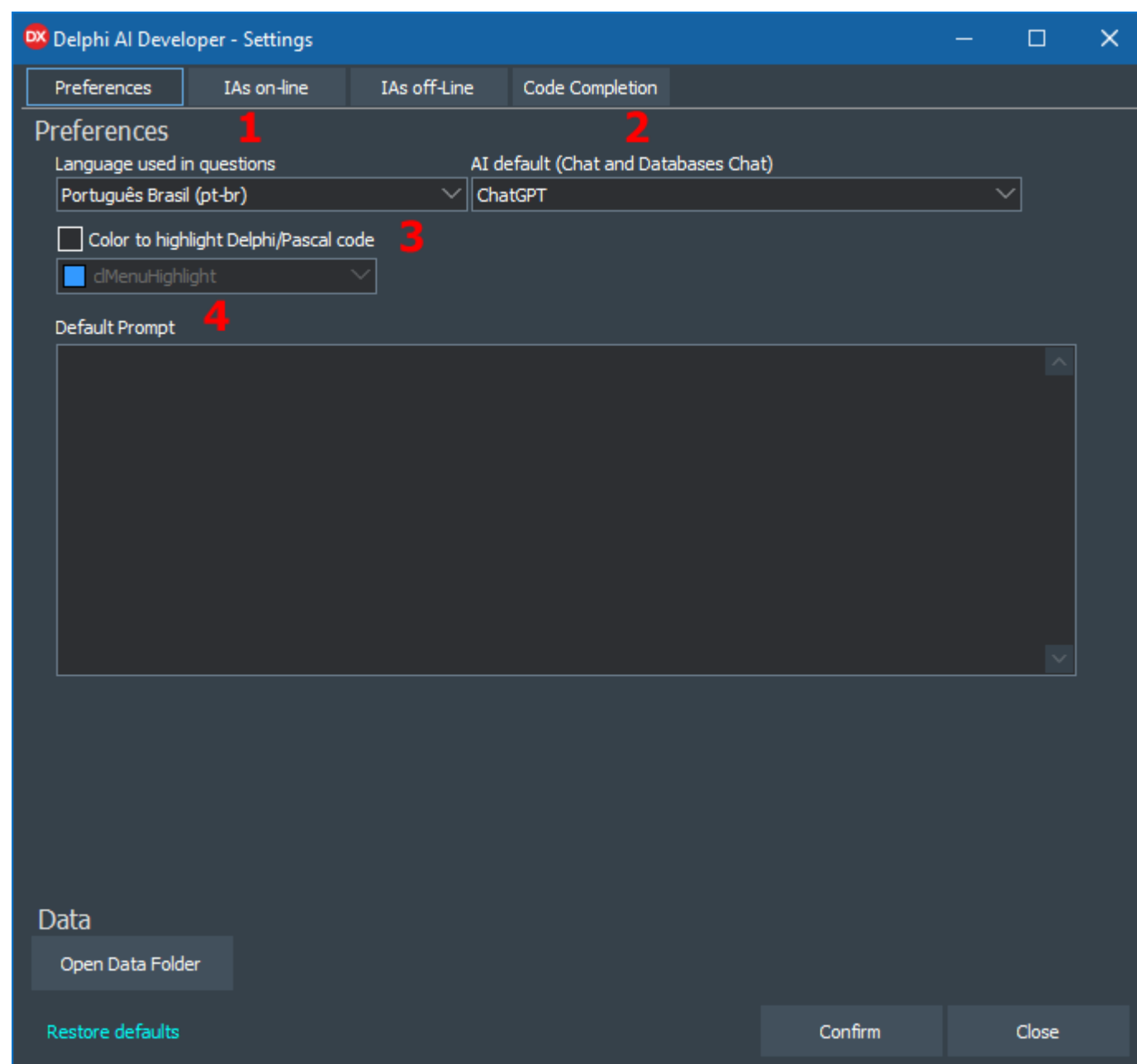
# Ollama-Delphi-GUI



<https://github.com/HuichanKIM/Ollama-Delphi-GUI>



# Delphi-AI-Developer

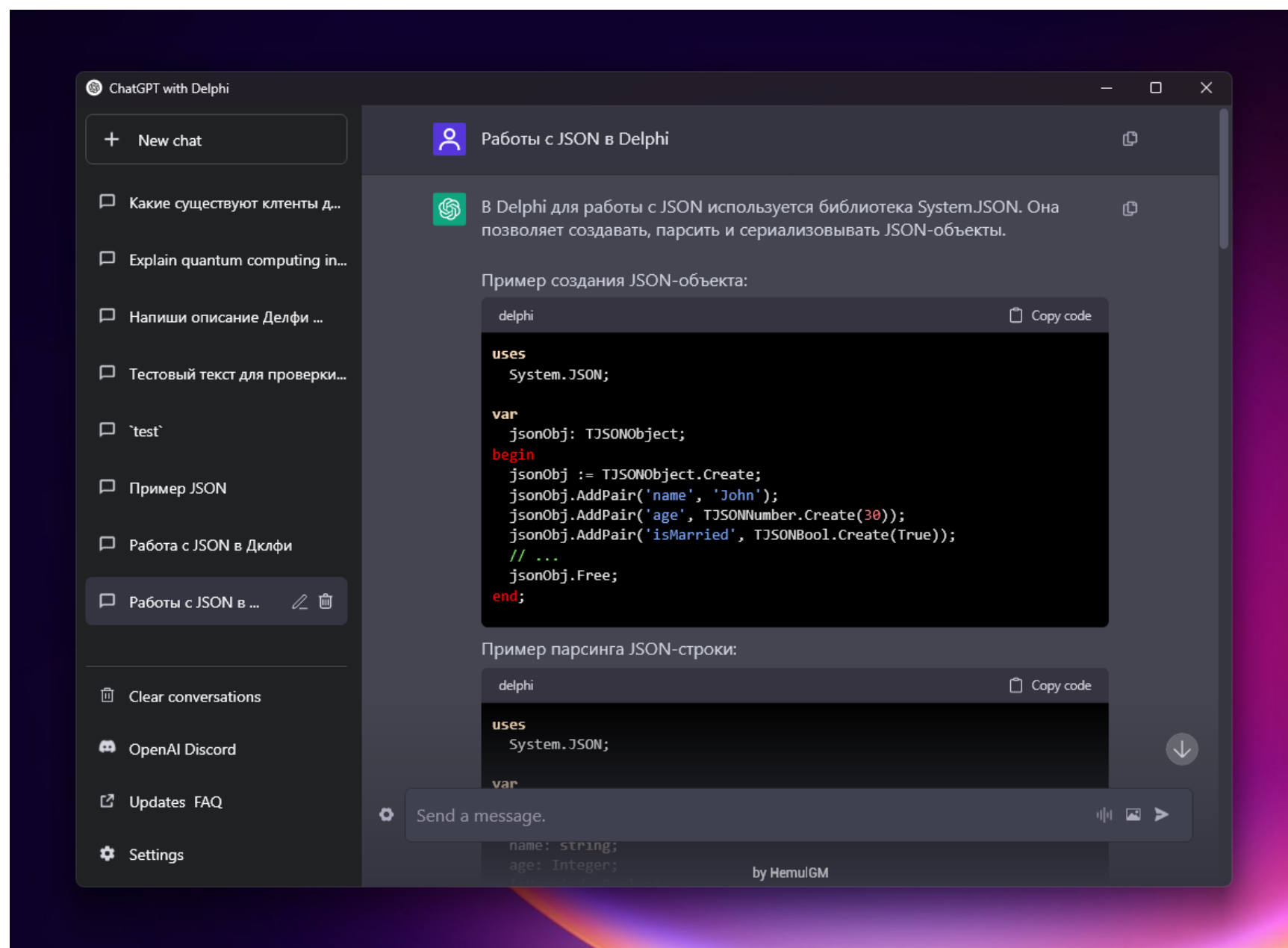


<https://github.com/Code4Delphi/Delphi-AI-Developer>





# Delphi OpenAI



<https://github.com/HemulGM/DelphiOpenAI>

# 할 수 있는 것, 바라는 것

## 할 수 있는 것

- 자연어 질의로 오브젝트 파스칼 코드 생성
- 버그 탐색, 코드 최적화 및 테스트 유닛 추가
- 코드 분석, 코드 완성, 주석 삽입
- 제한적인 폼 디자인 (코드를 기반으로)

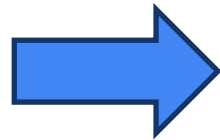
## 바라는 것

- 좀더 많은 학습을 기반으로 생성해주는 코드의 품질이 좋았으면...
- Visual Designer와의 통합



# 샘플 프로그램

## AI 를 활용하여 영수증 데이터 가져오기



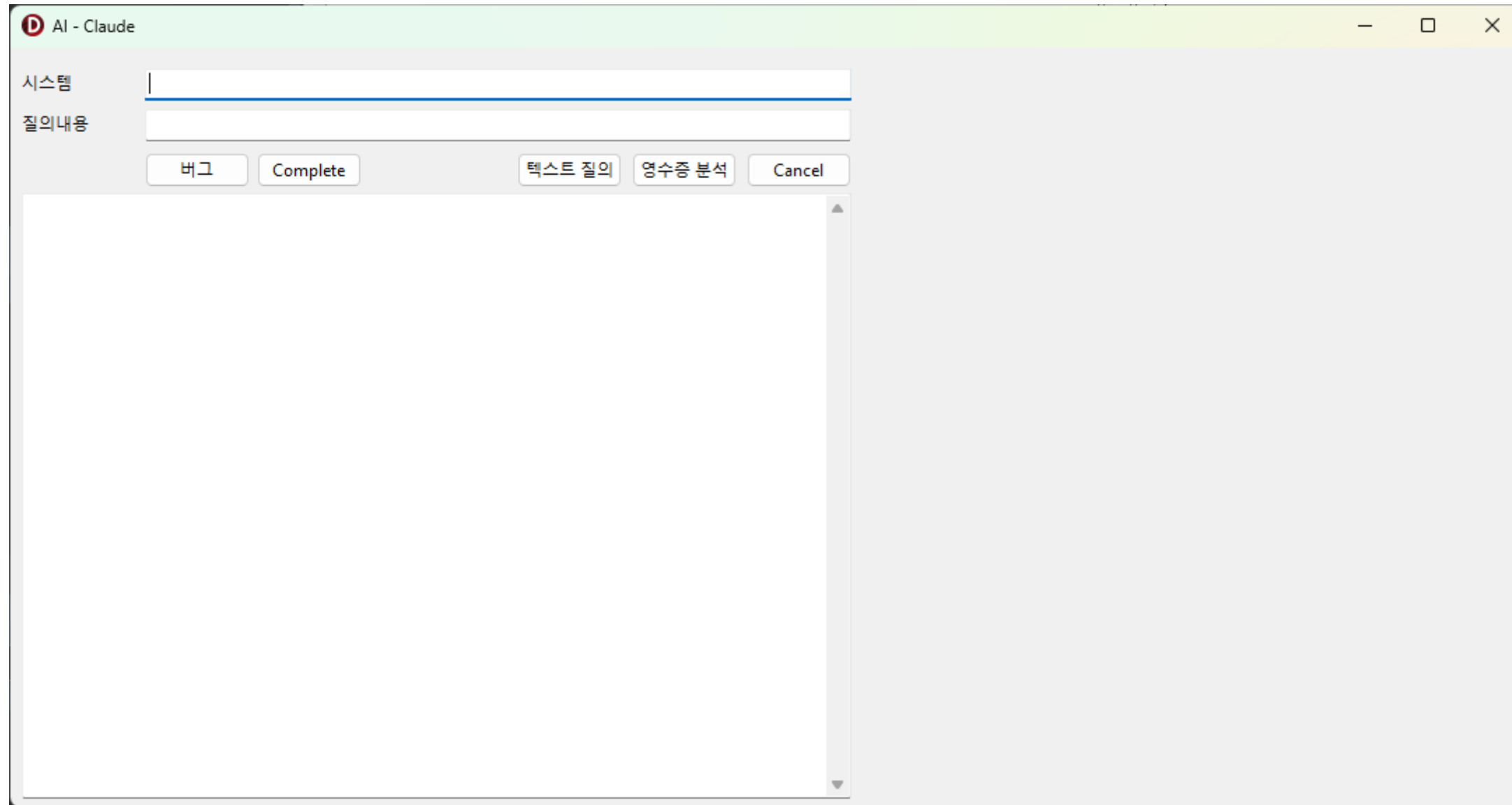
API 호출



결과 (Markdown)



# 샘플 프로그램 살펴보기







It's Delphi's  
**30<sup>th</sup>**!



