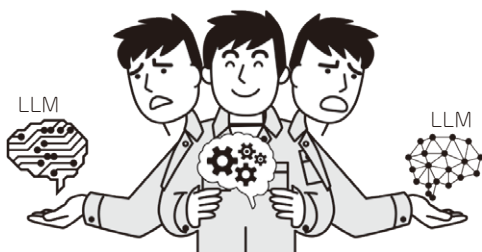


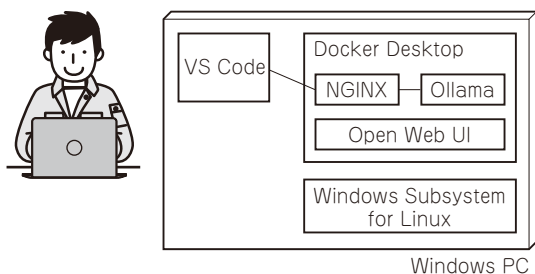
AIコーディング・アシスタント導入のススメ

ご購入はこちら

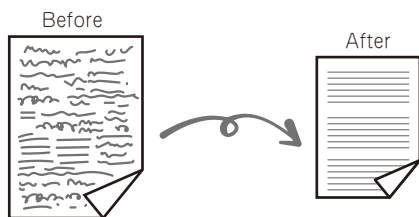
氏森 充



(a) 第2章…コーディングに利用できるLLMの比較実験



(b) 第3章…AIコーディング・アシスタントを作る



(c) 第4章…リファクタリングを試す

図1 本特設記事でやること

AIにコードを書かせたり、構造を解析させたりといった技術はAIコーディング・アシスタントとして存在感を高めており、今やソフトウェア開発のスタイルに大きな変革をもたらしつつあります。また、業務系アプリケーションから組み込み系、さらには研究開発まで、幅広い分野で注目が集まっています。

そこで特設では、次のことを解説、実験します。

第1章では、AIコーディング・アシスタントの導入を推奨する理由、およびその利点について説明しま

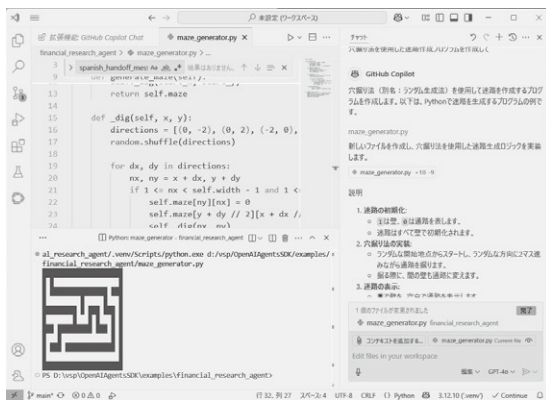


写真1 制作したAIコーディング・アシスタントを運用している様子

す。続く第2章では、第3章で実際に利用するLLM(大規模言語モデル)を選びます。第3章では、ローカル環境で動作するAIコーディング・アシスタントの作成手順を解説し、続く第4章では、作成したアシスタントを活用したリファクタリングの実例を紹介します(図1、写真1)。

AIコーディング・アシスタントの概要

AIコーディング・アシスタントは、ソフトウェア開発の生産性向上やコードの保守性改善を支援するツールとして、近年ますます注目を集めています。特に、初期実装から保守フェーズに至るまでの幅広い工程で有効に活用できる点が特徴です。その主な機能は、大きく分けて次の2つに分類されます。

● 機能①…コーディング支援

1つ目の機能は、コーディング支援です。これは次のように開発者の入力を補完/強化し、コーディングの手間を大幅に削減します。

- 関数名やコメントからのコード自動生成
- 文脈に応じた構文や変数名の補完
- エラー処理コードの自動挿入
- ログ出力コードの生成