

エンジニア向け生成AIの活用法

風間 正弘

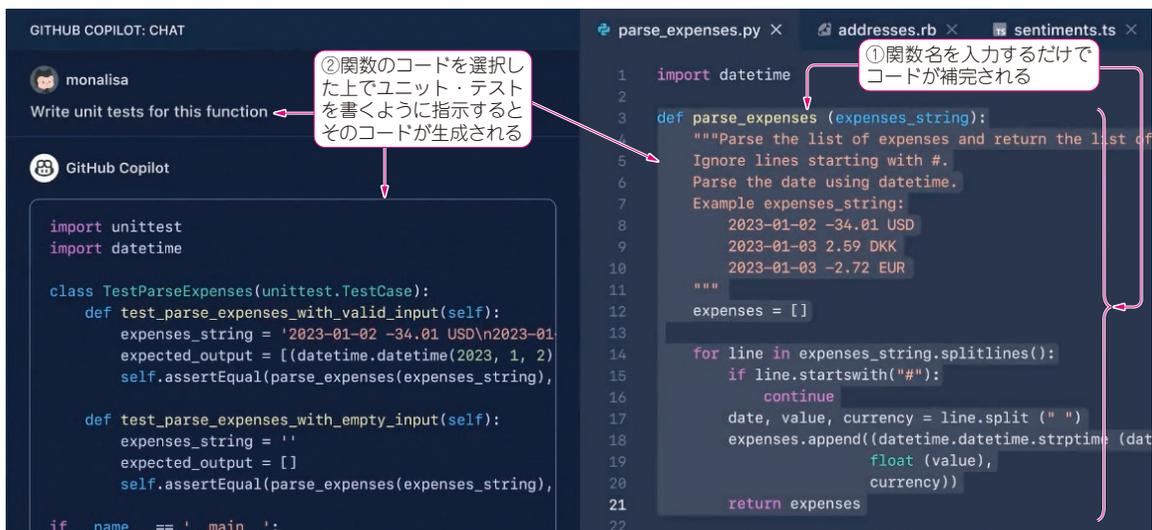


図1 関数名を入力するだけで、コードが自動で補完される関数内のコードおよびテスト用のコードが生成された状態

近年、人工知能の急速な進歩により、生成AIがさまざまな分野で利用されています。エンジニアリングの分野でも、生成AIを活用することで生産性の向上や品質の改善が期待できます。本稿では、生成AIを活用した生産性向上の事例について紹介します。

生成AIによるコーディング支援

エディタ上で生成AIを活用することで、コーディングの効率を向上させることができます。代表的なものとして、GitHub CopilotやCursorが挙げられます。

● GitHub Copilot

GitHub Copilotは、AIを活用したコーディング支援サービスです。GitHub上に存在する大量のコードを学習したAIを搭載しています。GitHub Copilotは、JetBrains IDE、Neovim、Visual Studio、Visual Studio Code (VSCode) などの各種エディタで利用できます。

コードの自動補完や関数名の提案など、コーディングを支援する機能を備えています。コードを書いている

と、次々とコードのサジェストが出てくるので、そこから選択することで、数千行のコードを効率良く作成できます。また、チャット欄で対話しながら、関数の実装を提案させられるなど、高度な機能も利用できます。

国内でも、IT企業を中心にGitHub Copilotを導入し、生産性が数十%向上した事例が報告されています。

▶使用例：Copilot

GitHub Copilotのリポジトリ⁽¹⁾では具体的な使用例が英語で紹介されています。

●コードの自動補完

わずかな情報を元に、コードが自動補完されます。図1では、`parse_expenses` (費用の抽出) という関数名を1行入力するだけで、関数内の数十行のコードが作成されています。

●テスト用コードの生成

関数のテスト用のコードを書きたい場合には、関数のコードを選択した上で、左のチャット欄に、「Write unit tests for this function (ユニット・テストのコードを書いて)」と記入するとテスト用のコードを書いてくれます(図1)。チャット欄を活用することで、エ