Llama-copilot VSCode Extension Front-end Design: Classes/Modules and Interfaces

The extension allows users to paste code snippets into a chat box within a webview.

Module 1: `extension.ts`

This is the main module of the VSCode extension, responsible for activating the extension, registering commands, and handling interactions between the editor and the webview.

Public Interfaces:

- **`activate(context: vscode.ExtensionContext)`**
  This function is executed when the extension is activated. It registers commands, creates the webview panel, and listens for text selections and message passing.

- **`deactivate()`**
  This function is called when the extension is deactivated to clean up resources.

Key Interaction Methods:

- **`vscode.commands.registerCommand('chatbox.open', () => {...})`**
  Registers and defines the command `chatbox.open`. When triggered, this command opens the chat box webview.

- **`vscode.window.onDidChangeTextEditorSelection(event => {...})`**
  Listens to text selections in the editor and copies the selected text to the clipboard.

- **`panel.webview.onDidReceiveMessage(message => {...})`**
  Listens for messages from the webview and handles requests to paste code.

- **`pasteCodeToChatBox(panel: vscode.WebviewPanel)`**
  Reads code from the clipboard and sends it to the webview to be displayed in the chat box.

Module 2: `webviewContent.ts`

This module generates the HTML and CSS content for the webview, defining the structure and style of the chat box, and implements the JavaScript logic for interacting with the chat box.

Public Interface:

- **`getWebviewContent()`**
  This function returns an HTML string that contains the structure, styles, and JavaScript logic for the chat box in the webview.

Key Interaction Functions:

- `sendMessage()`
  Sends the user input from the chat box and displays it in the webview.

- `document.addEventListener('contextmenu', event => {...})`
  Listens for right-click events and triggers the paste code request.

- `window.addEventListener('message', event => {...})`
  Listens for messages sent from the extension and handles the paste code operation.

Summary

The extension consists of two main modules:

1. `extension.ts`: Handles the activation of the extension, command registration, and interaction with the editor.
2. `webviewContent.ts`: Generates the HTML content for the webview and manages interactions within the webview.

The code is structured to allow seamless communication between the VSCode editor and the chat box webview, enabling the user to paste code snippets directly into the chat interface.