**Descrição Geral**

A classe LoggerSetup é projetada para facilitar a configuração e utilização de logs em uma aplicação. Ela define a configuração básica para a criação de logs em um arquivo, permitindo registrar mensagens em diferentes níveis de severidade, como debug, info, warning e error.

**Bibliotecas Importadas**

* **logging**: Usada para criar logs de diferentes níveis de severidade.
* **sys**: Fornece acesso a variáveis e funções específicas do sistema.
* **os**: Usada para manipulação de caminhos de arquivos e diretórios no sistema operacional.

**Classe LoggerSetup**

**Inicialização**

* **Método \_\_init\_\_**: Configura os atributos iniciais necessários para criar um logger.
  + **Parâmetros**:
    - pathLog: Caminho onde o arquivo de log será armazenado.
    - nameLog: Nome do arquivo de log.
    - nameLogger: Nome do logger, que será utilizado para identificar as mensagens de log.

**Métodos**

1. **\_setup\_logger**:
   * **Propósito**: Configura e retorna um objeto logger com as definições de formato e nível especificados.
   * **Parâmetros**: Recebe o nível de log desejado como uma string (levellOG).
   * **Funcionamento**:
     + Configura o formato padrão para as mensagens de log.
     + Cria e configura um formatter para definir como as mensagens de log serão formatadas.
     + Determina o caminho completo onde o arquivo de log será armazenado.
     + Cria um logger com o nome fornecido no momento da inicialização.
     + Configura um manipulador de arquivos (FileHandler) para gravar logs em um arquivo.
     + Define o nível de log do manipulador baseado no parâmetro fornecido.
     + Adiciona o manipulador ao logger.
     + Retorna o objeto logger configurado.
2. **debugMessage**:
   * **Propósito**: Registra uma mensagem de nível debug no arquivo de log.
   * **Parâmetros**:
     + message: Mensagem a ser registrada.
     + level: Nível de log como string, que determina a severidade mínima para a mensagem ser registrada.
   * **Funcionamento**: Chama o método \_setup\_logger para configurar o logger e registra a mensagem de nível debug.
3. **infoMessage**:
   * **Propósito**: Registra uma mensagem de nível info no arquivo de log.
   * **Parâmetros**:
     + message: Mensagem a ser registrada.
     + level: Nível de log como string, que determina a severidade mínima para a mensagem ser registrada.
   * **Funcionamento**: Chama o método \_setup\_logger para configurar o logger e registra a mensagem de nível info.
4. **errorMessage**:
   * **Propósito**: Registra uma mensagem de nível error no arquivo de log.
   * **Parâmetros**:
     + message: Mensagem a ser registrada.
     + level: Nível de log como string, que determina a severidade mínima para a mensagem ser registrada.
   * **Funcionamento**: Chama o método \_setup\_logger para configurar o logger e registra a mensagem de nível error.
5. **warningMessage**:
   * **Propósito**: Registra uma mensagem de nível warning no arquivo de log.
   * **Parâmetros**:
     + message: Mensagem a ser registrada.
     + level: Nível de log como string, que determina a severidade mínima para a mensagem ser registrada.
   * **Funcionamento**: Chama o método \_setup\_logger para configurar o logger e registra a mensagem de nível warning.

**Resumo**

A classe LoggerSetup facilita a criação e uso de logs em uma aplicação, permitindo a configuração flexível de níveis de severidade e formatos de mensagem. Isso é útil para monitorar e registrar o comportamento do aplicativo em arquivos de log, ajudando na identificação e resolução de problemas.