**Asynchrone Datenverarbeitung mit async/await**

**Hintergrund:**  
Sie sollen eine kleine Anwendung erstellen, die Daten aus mehreren Quellen gleichzeitig abruft und verarbeitet. Ziel ist es, die Vorteile der asynchronen Programmierung zu nutzen, um die Wartezeit auf langsame Datenquellen zu minimieren.

**Aufgabenbeschreibung:**

1. **Datenquellen simulieren:**  
   Erstellen Sie vier Methoden, die jeweils Daten „abrufen“ und dabei unterschiedliche Wartezeiten simulieren:

|  |  |  |
| --- | --- | --- |
| Methode | Rückgabewert | Wartezeit |
| Task<string> GetUserDataAsync() | string | 2 Sek. |
| Task<int> GetUserAgeAsync() | int | 1 Sek. |
| Task<List<string>> GetProductListAsync() | List<string> | 3 Sek. |
| Task<double> GetOrderTotalAsync() | double | 1,5 Sek. |

1. Jede Methode soll am Ende den entsprechenden Wert zurückgeben, z. B. "Benutzerdaten abgerufen" oder 42.
2. **Asynchrone Verarbeitung:**
   * Schreiben Sie eine Methode Task LoadAllDataAsync(), die alle vier Datenquellen **parallel** abruft.
   * Geben Sie die Ergebnisse nacheinander auf der Konsole aus.
   * Nutzen Sie async/await korrekt, sodass die Methoden nicht blockierend sind.
3. **Bonusaufgabe:**
   * Fügen Sie eine Methode Task<string> GetDataWithErrorAsync() hinzu, die zufällig eine Exception wirft.
   * Erweitern Sie Ihre asynchrone Methode, sodass Fehler abgefangen und auf der Konsole protokolliert werden, ohne dass die Anwendung abstürzt.