**Dokumentation der SearchTracker-Anwendung**

**1. Einführung**

Die **SearchTracker**-Anwendung ist eine C#-basierte Desktop-Applikation, die es Nutzern ermöglicht, das Erscheinungsbild der Benutzeroberfläche zwischen den zwei Themen "Hell" und "Dunkel" umzuschalten. Sie verwendet Standard-Windows-Forms, um ein einfaches und konsistentes UI-Design zu bieten. Die Anwendung ist modular aufgebaut und umfasst mehrere Hauptkomponenten, die jeweils spezifische Funktionen und Zuständigkeiten haben.

**2. Architektur und Hauptkomponenten**

Die Anwendung ist in verschiedene Klassen aufgeteilt, die jeweils unterschiedliche Aspekte der Anwendung verwalten:

1. **Program.cs**:
   * **Funktion**: Dient als Haupteinstiegspunkt der Anwendung.
   * **Beschreibung**: Initialisiert die Anwendung, startet die Logger-Session und startet das Hauptfenster.
2. **MainWindow.cs**:
   * **Funktion**: Hauptfenster der Anwendung, das die Benutzeroberfläche bereitstellt.
   * **Beschreibung**: Verwaltet die Benutzerinteraktionen und bietet Zugriff auf das Fenster zur Theme-Verwaltung. Enthält Methoden zur Anwendung des ausgewählten Themas.
3. **ThemeManageWindow.cs**:
   * **Funktion**: Fenster zur Verwaltung der Theme-Einstellungen.
   * **Beschreibung**: Bietet eine Benutzeroberfläche zur Auswahl des UI-Themas. Die Auswahl wird gespeichert und auf die Anwendung angewendet.
4. **ThemeManager.cs**:
   * **Funktion**: Statische Klasse zur Verwaltung der UI-Themen.
   * **Beschreibung**: Enthält Methoden zur Anwendung von Themes ("Hell" und "Dunkel") und zum Einstellen der Schriftfarben. Verwendet zur Laufzeit aktuelle UI-Einstellungen auf alle offenen Formulare.
5. **Logger.cs**:
   * **Funktion**: Statische Klasse für das Protokollieren von Debug-Nachrichten.
   * **Beschreibung**: Ermöglicht das Starten und Beenden von Log-Sessions, sowie das Aufzeichnen von Ereignissen und Fehlern in einer Log-Datei.
6. **MainWindow.Designer.cs**:
   * **Funktion**: Automatisch generierte Datei, die die UI-Komponenten des Hauptfensters definiert.
   * **Beschreibung**: Enthält die Definition der Menüstruktur und anderer UI-Komponenten.
7. **ThemeManageWindow.Designer.cs**:
   * **Funktion**: Automatisch generierte Datei, die die UI-Komponenten des Theme-Management-Fensters definiert.
   * **Beschreibung**: Enthält die Definition der ComboBox zur Auswahl des Themas und zugehöriger Labels.

**3. Themenmanagement**

Die Anwendung bietet zwei Hauptthemen: "Hell" und "Dunkel". Diese Themen ändern die Hintergrund- und Schriftfarben der Anwendung, um eine anpassbare Benutzererfahrung zu bieten:

* **Hell**: Standardfarben, die für eine gut beleuchtete Umgebung geeignet sind.
* **Dunkel**: Dämpfere Farben mit weißen Schriftfarben, ideal für dunkle Umgebungen oder um die Belastung der Augen zu verringern.

**4. Debugging und Logging**

Die **Logger**-Klasse wird verwendet, um alle wichtigen Ereignisse und Fehler zu protokollieren. Diese Protokolle helfen bei der Überwachung und Fehlerbehebung der Anwendung. Die Nachrichten sind klar strukturiert und enthalten:

* Start- und Endmeldungen für die Anwendung.
* Meldungen beim Wechsel der UI-Themen.
* Debug-Meldungen für Benutzeraktionen wie das Öffnen des Theme-Management-Fensters.

**5. Benutzerschnittstelle und Interaktion**

Die Benutzeroberfläche der Anwendung ist einfach und intuitiv. Sie besteht aus einem Hauptmenü, das Optionen zum Ändern des UI-Themas und zum Beenden der Anwendung bietet. Das Theme-Management-Fenster ermöglicht die Auswahl zwischen den verfügbaren Themen und zeigt die aktuelle Auswahl an.

**6. Erweiterungsmöglichkeiten**

Die Anwendung ist so gestaltet, dass sie leicht erweitert werden kann. Zum Beispiel können neue Themen hinzugefügt oder zusätzliche Funktionen zur UI-Anpassung implementiert werden. Die **ThemeManager**-Klasse bietet eine zentrale Stelle, an der solche Erweiterungen vorgenommen werden können.

**7. Fazit**

Die SearchTracker-Anwendung bietet eine solide Grundlage für eine anpassbare Benutzeroberfläche in einer Windows-Forms-Umgebung. Die klare Trennung der Verantwortlichkeiten in verschiedene Klassen erleichtert die Wartung und Erweiterung der Anwendung. Dank der umfassenden Logging-Funktionalität können Benutzer und Entwickler die Anwendung einfach überwachen und potenzielle Probleme diagnostizieren.