**Logging / Excepton Handling**

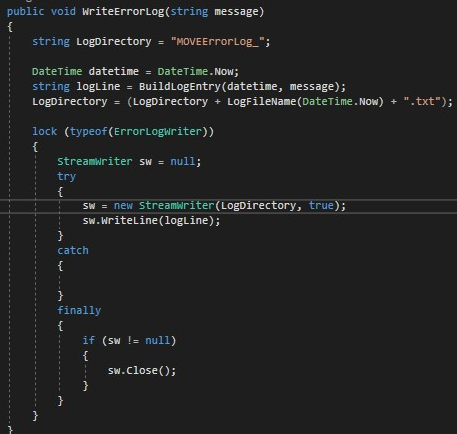
In unserem Projekt wurde eine Loggingklasse implementiert, welche Aufzeichnungen zu sämtlichen Fehlern in unserem Projekt aufzeichnen soll, um diese zu gegebener Zeit analysieren zu können. Im Falle eines unerwarteten Fehlers wird durch richtiges Exception Handling die Loggingmethode aufgerufen und kommt zum Einsatz. Die Softwareanwendung läuft wie zuvor ohne Absturz weiter. Für jeden Tag wird eine neue ErrorLog-Datei angelegt, um eine bessere Übersicht zu bekommen. (Abbildung 1.1).



Abbildung 1.1

**Code**

In der Methode „WriteErrorLog“ (Abbildung 1.1), wird zuerst ein Standort dieser Logdatei festgelegt, wohin das Programm die entstandenen Fehlermeldungen abspeichern soll. In unserem Fall ist dieser Standort direkt im Debug Verzeichnis des Softwareprojektes mit einem selbsternannten Namen, dem derzeitigen Datum. Jeder neuer Eintrag wird „gelockt“ bevor es zum tatsächlichen Eintrag kommt, dies hat den Grund, dass kein anderer Thread gleichzeitig eine Fehlermeldung schreiben kann, sondern dieser Thread warten muss bis die Sperre aufgehoben wird bzw. die Fehlermeldung in die Logdatei reingeschrieben wurde. Die Lock-Anweisung bewirkt die gegenseitige Ausschlusssperre für ein bestimmtes Objekt, in unserem Fall das Objekt „ErrorLogWriter“.



Die Methode BuildLogEntry in Abbildung 1.2, erstellt den Logeintrag mit einem gewissen Format. Das Format (Abbildung 1.3) besteht aus dem genauen Zeitpunkt des Ereignisses (dd-MM-yyyy HH:mm:ss), danach folgt der Name der Exception mit der entsprechenden Nachricht dieser Exception.

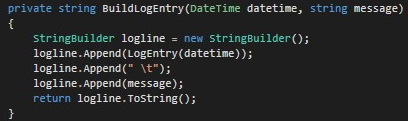


Abbildung 1.2



Abbildung 1.3

Die Methoden „LogEntry“ und “LogFileName“ in Abbildung 1.1 werden für die jeweiligen zwei verschiedenen Formate des Datums verwendet zum einen die exakte Zeit für den Logeintrag, zum anderen nur das Datum für das neue Anlegen einer Logdatei.

