



## AccountManager, Account, Payment

- Der **AccountManager** verwaltet eine Menge von **Accounts**. Ein Account hat eine (gültige) Kreditkartennummer und weiters kann einem **Account** eine Menge von **Payments** zugewiesen sein.
- Die Datei Payments.csv enthält einzelne **Payments**, beim Einlesen dieser Datei soll ein **Dictionary** aufgebaut werden, in dem einer gültigen Kreditkartennummer genau ein **Account** zugeordnet ist. Jedes der eingelesenen **Payments** ist dann entweder einem bereits bestehenden **Account** oder einem gerade erst erzeugten **Account** zuzuordnen.
- Ein **Account** darf nur mit einer gültigen Kreditkartennummer angelegt werden, andernfalls wird eine **ArgumentException** erzeugt.

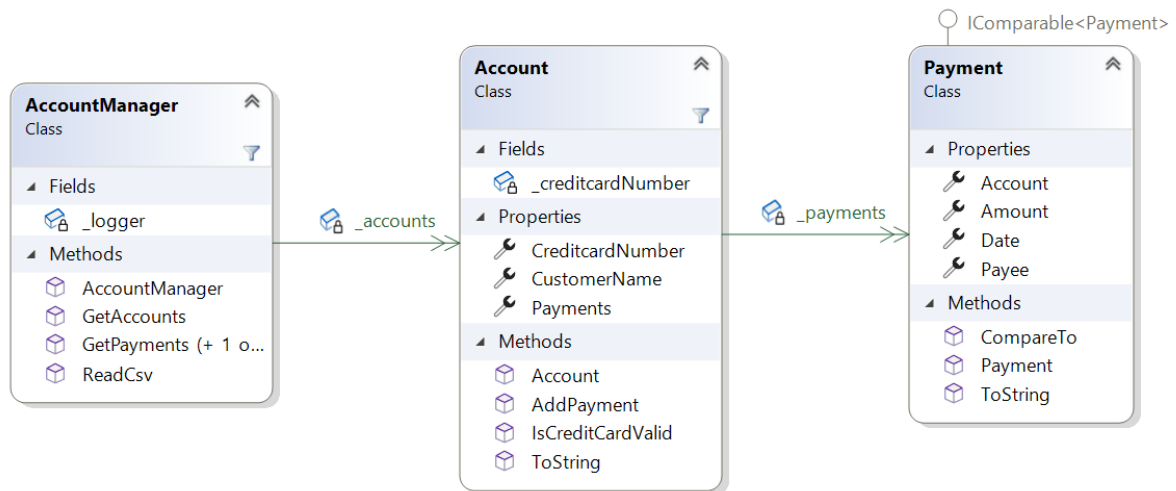
## AccountManager – öffentliche Methoden

- **ReadCsv**: Einlesen der csv-Datei, Aufbau des Dictionary
  - o Parameter **fileName**
  - o Parameter **continueOnError**
    - **True**: Ungültige Kreditkartennummern werden im Logger als Error protokolliert und nicht weiter behandelt. Das Einlesen der Payments wird fortgesetzt.
    - **False**: Andernfalls wird die Verarbeitung der Datei abgebrochen, indem die aufgetretene Exception an die oberste Schicht der Anwendung weitergegeben wird.
- **GetAccounts**
  - o Der AccountManager liefert eine unsortierte Liste bestehend aus allen Accounts  
(`List<Account>`)
- **GetPayments**
  - o Liefern eines sortierten IEnumerable von Payments, deren Empfängernamen entsprechend anfängt
  - o Liefern eines sortierten IEnumerable von Payments, die an einem bestimmten Datum stattgefunden haben
  - o Die Sortierung der Payments erfolgt nach Zahlungsdatum.

```
/// <summary>
/// Alle Zahlungen an Empfänger, deren Name entsprechend anfängt, zurückgeben
/// </summary>
/// <param name="payeeStartsWith">Empfängername</param>
/// <returns>Betreffende Zahlungen, sortiert nach Datum</returns>
public IEnumerable<Payment> GetPayments(string payeeStartsWith)
```

```
/// <summary>
/// Alle Zahlungen an dem Tag zurückgeben
/// </summary>
/// <param name="date">Selektiertes Datum</param>
/// <returns>Zahlungen an dem Tag</returns>
public IEnumerable<Payment> GetPayments(DateTime date)
```

## Vorschlag für ein Klassendiagramm



## Fehlerbehandlung, Logging

Der AccountManager verlangt im Konstruktor ein ILogger-Objekt, um entsprechend dem Parameter `continueOnError` in der Methode `ReadCsv` die passende Fehlerbehandlung durchführen zu können.

Ein konkreter ILogger bietet Implementierungen für das Logging von Errors und Infos an. Es sind drei konkrete ILogger-Implementierungen vorzusehen und wahlweise über den AccountManager zu testen:

- **ConsoleLogger:**
  - o Ausgabe von Errors mit roter ForegroundColor und dem Präfix „ERROR: „
  - o Infos werden in gelb und mit dem Präfix „INFO: „ auf die Konsole geschrieben.
- **FileLogger:**
  - o Im Konstruktor ist ein Dateiname anzugeben, die Ausgabe erfolgt im CSV-Format
  - o Errors: 1. Spalter „ERROR“, 2. Spalte Datum und Uhrzeit, 3. Spalte: Konkrete Fehlermeldung
  - o Infos: 1. Spalte „INFO“, 2. Spalte Datum und Uhrzeit, 3. Spalte: Info-Text
- **LoggerCollection:**
  - o Dieser Logger nimmt im Konstruktor eine beliebige Menge von ILogger-Instanzen auf
  - o LogError und LogInfo geben die konkrete Log-Nachricht an alle Instanzen der Collection weiter.
- Mit dem Property `LogLevel` kann man einstellen, ob ein Logger nur Errors, oder Errors und Infos ausgeben soll (wenn man INFOS sehen will, will man sicher auch ERRORS sehen ...)
- Beispiel zum gleichzeitigen Loggen in eine Log-Datei und auf die Konsole, wobei die Konsole nur Errors ausgibt, in der Datei werden aber Errors und Infos geloggt:

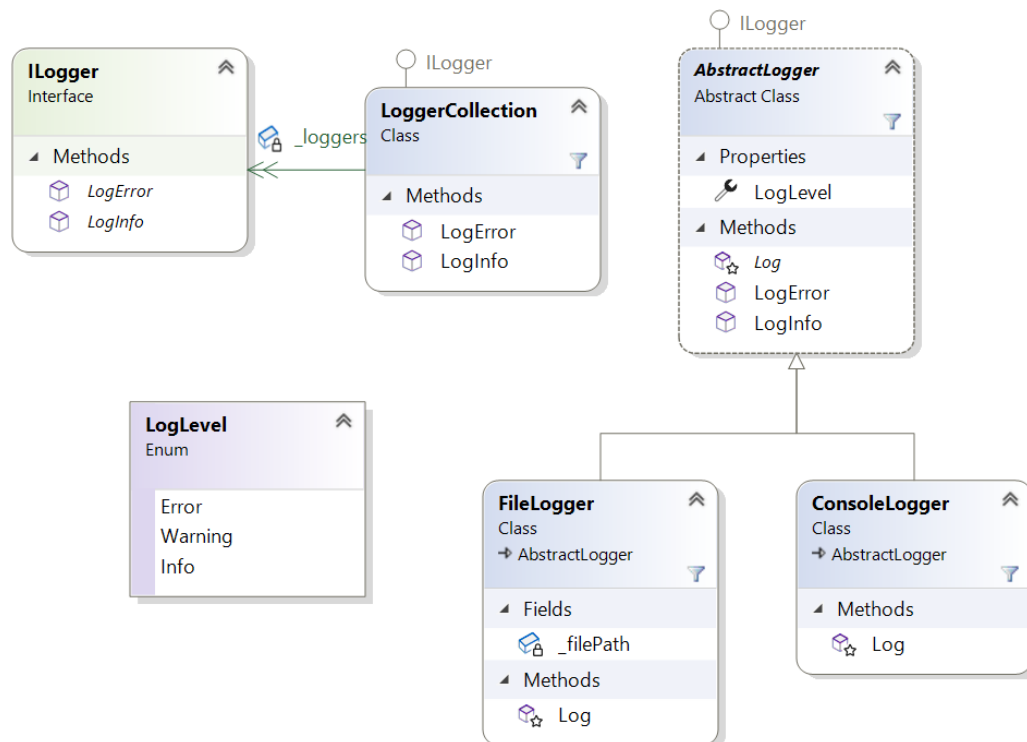
```

ILogger consoleLogger = new ConsoleLogger(LogLevel.Error);
ILogger fileLogger = new FileLogger("log.csv", LogLevel.Info);
ILogger logger = new LoggerCollection(fileLogger, consoleLogger);
AccountManager accounts = new AccountManager(logger);
  
```

```
>>ERROR:      Kreditkartennummer 7875259492246491 ist ungültig
>>ERROR:      Kreditkartennummer 7875259492246491 ist ungültig
>>ERROR:      Kreditkartennummer 1429670339571291 ist ungültig
>>ERROR:      Kreditkartennummer 1429670339571291 ist ungültig
```

log.csv

```
1 INFO; 30.03.2017 18:16:38;Account 1771914536488775 angelegt.
2 INFO; 30.03.2017 18:16:38;Account 6288081682946785 angelegt.
3 ERROR; 30.03.2017 18:16:39;Kreditkartennummer 7875259492246491 ist ungültig
4 INFO; 30.03.2017 18:16:39;Account 7341247641887864 angelegt.
5 ERROR; 30.03.2017 18:16:39;Kreditkartennummer 7875259492246491 ist ungültig
6 INFO; 30.03.2017 18:16:39;Account 9256328210666928 angelegt.
7 INFO; 30.03.2017 18:16:39;Account 9096011694165260 angelegt.
```



## Hauptprogramm

Das Hauptprogramm nimmt folgende Kommandozeilenparameter entgegen:

- -ce true|false : Setzt "continueOnError" auf true oder false. Verpflichtend.
- -p paymentsFile : voller Pfad zu dem CSV file. Optional. Default: Payments.csv
- -l logFile : voller Pfad zu einem Logfile. Muss Dateierweiterung haben (z.B. "log.txt"). Optional. Default: log.txt

Wenn continueOnError false ist, bricht das Einlesen der CSV-Dateien beim ersten fehlerhaften Eintrag ab.

```
INFO>> Es wurde die Option 'continueOnError=False' gewählt.
Creditcard Clearing
=====
INFO>> Account 1771914536488775 angelegt.
INFO>> Account 6288081682946785 angelegt.
ERROR>> Abbruch Einlesen Payments!
Kreditkartennummer 7875259492246491 ist ungültig
```

Falls irgendetwas schiefgeht (z.B. Dateien sind blockiert oder werden nicht gefunden), oder z.B. continueOnError auf false gestellt wird (= kein Kommandozeilenargument gesetzt), so hat das Hauptprogramm auf den jeweiligen Fehler zu reagieren (Ausgabe Fehlermeldung) und danach wird das Programm sauber niedergefahren.

```
INFO>> Es wurde die Option 'continueOnError=True' gewählt.
Creditcard Clearing
=====
INFO>> Account 1771914536488775 angelegt.
INFO>> Account 6288081682946785 angelegt.
ERROR>> Kreditkartennummer 7875259492246491 ist ungültig
INFO>> Account 7341247641887864 angelegt.
ERROR>> Kreditkartennummer 7875259492246491 ist ungültig
INFO>> Account 9256328210666928 angelegt.
INFO>> Account 9096011694165260 angelegt.
INFO>> Account 4861749166865115 angelegt.
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
INFO>> Account 0380279935255026 angelegt.
INFO>> Account 0896388906596503 angelegt.
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
ERROR>> Kreditkartennummer 7875259492246491 ist ungültig
ERROR>> Kreditkartennummer 7875259492246491 ist ungültig
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
ERROR>> Kreditkartennummer 7875259492246491 ist ungültig
ERROR>> Kreditkartennummer 7875259492246491 ist ungültig
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
ERROR>> Kreditkartennummer 7875259492246491 ist ungültig
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
ERROR>> Kreditkartennummer 7875259492246491 ist ungültig
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
ERROR>> Kreditkartennummer 7875259492246491 ist ungültig
ERROR>> Kreditkartennummer 1429670339571291 ist ungültig
78 Zahlungen für 8 Konten eingelesen.
```

Falls das Einlesen der Daten von der CSV-Datei fortgesetzt wird, werden folgende Ausgaben erwartet:

78 Zahlungen für 8 Konten eingelesen.

Kontenübersicht

Kunde: Michalik	Kreditkarte: 1771914536488775
Kunde: Mayer	Kreditkarte: 6288081682946785
Kunde: Mally	Kreditkarte: 7341247641887864
Kunde: Niedermayr	Kreditkarte: 9256328210666928
Kunde: Pernsteiner	Kreditkarte: 9096011694165260
Kunde: Schimbäck	Kreditkarte: 4861749166865115
Kunde: Pöschko	Kreditkarte: 0380279935255026
Kunde: Kulig	Kreditkarte: 0896388906596503

Zahlungen an Mediamarkt

Kunde: Kulig	Kreditkarte: 0896388906596503	Zahlung an MediaMarkt	04.10.2008:	953,04
Kunde: Mally	Kreditkarte: 7341247641887864	Zahlung an MediaMarkt	01.12.2008:	304,29
Kunde: Niedermayr	Kreditkarte: 9256328210666928	Zahlung an MediaMarkt	30.06.2009:	778,13
Kunde: Mayer	Kreditkarte: 6288081682946785	Zahlung an MediaMarkt	18.10.2009:	150,84
Kunde: Pernsteiner	Kreditkarte: 9096011694165260	Zahlung an MediaMarkt	10.01.2010:	221,32
Kunde: Michalik	Kreditkarte: 1771914536488775	Zahlung an MediaMarkt	04.02.2010:	194,91

Zahlungen am 28.9.2008

Kunde: Mally	Kreditkarte: 7341247641887864	Zahlung an Fussl	28.09.2008:	588,89
Kunde: Michalik	Kreditkarte: 1771914536488775	Zahlung an Dm	28.09.2008:	830,56

Danach beendet das Programm.

Im Sourcecode sind einige TODO-Einträge hinterlegt, welche alle bearbeitet werden müssen!