

Protokoll „Datei-Komprimierung“

Von Simon Bullik, DQI16

Schulzentrum Utbremen

SZ SII Utbremen	DQI16	Stand: 17.12.2017
PP Gruppe 1	Simon Bullik	Jürgen Wolkenhauer
	Protokoll „Datei-Komprimierung“	

Inhalt

Aufgabenstellung	3
Ausstattung.....	3
Bei Programmen:	3
Benutzeranleitung.....	3
Funktionsbeschreibung.....	3
Erläuterung der einzelnen Programmteile	3
Softwaremodule erläutern.....	3
Zusammenarbeit der Klassen.....	4
Programmtests	4
Erkenntnisgewinn.....	4

SZ SII Utbremen	DQI16	Stand: 17.12.2017
PP Gruppe 1	Simon Bullik	Jürgen Wolkenhauer
	Protokoll „Datei-Komprimierung“	

Aufgabenstellung

Die Aufgabe war ein Programm zur Reduzierung der Größe von Dateien zu erstellen. Dies sollte nach den beiden beschriebenen Algorithmen mit dem Prinzip des „Runlength-Encoders“ geschehen. Außerdem soll das Programm noch Parameter in der Konsole aufnehmen.

Ausstattung

Es wird ein Endgerät mit einem installiertem Python3 Interpreter benötigt.

Bei Programmen:

Benutzeranleitung

Zum Benutzen müssen Sie die Datei *start.py* mit den gewünschten Parametern starten. Möglich Aufrufe wären zum Beispiel:

```
python start.py -c/-u -f <test.bmp> [-e rl3] [-q]
```

```
python start.py --compress/--uncompress --file=<test.bmp> [--extension=rl3] [--quiet]
```

Die Werte müssen natürlich angepasst werden.

Funktionsbeschreibung

Das Programm kann über die Konsole Dateien mit dem Algorithmus RL3 komprimieren und Dateien dekomprimieren, welche mit RL2 oder RL3 komprimiert worden sind. Des Weiteren kann man eine eigene Dateiendung für die komprimierte Datei festlegen und auswählen, ob Ausgaben in der Konsole erfolgen sollen.

Erläuterung der einzelnen Programmteile

Softwaremodule erläutern

Start.py

In der Datei *start.py* findet die Parameterverarbeitung mit *argparse* statt.

RLedInterface.py

Hier befindet sich das Grundgerüst der beiden Algorithmen. Es enthält die beiden Methoden *compress* und *expand*, welche ohne Funktion sind. Des Weiteren enthält es auch die Attribute mit

SZ SII Utbremen	DQI16	Stand: 17.12.2017
PP Gruppe 1	Simon Bullik	Jürgen Wolkenhauer
	Protokoll „Datei-Komprimierung“	

der maximalen Anzahl von Bytes welche komprimiert werden können und die Anzahl der Bytes, welche auf einmal eingelesen werden.

[RL2.py](#)

Hier ist der erste (einfache) Algorithmus implementiert.

[RL3.py](#)

Hier ist der zweite Algorithmus implementiert.

[RLedSchnittstelle.py](#)

Das ist die Schnittstelle, welche uns zur Verfügung gestellt worden ist. Hier werden in den Methoden jeweils ein Objekt von dem jeweiligen Algorithmus erzeugt und die jeweilige Aktion ausgeführt. Hier kann eingestellt werden, ob Fehler geworfen werden sollen oder ob das Programm still beendet werden soll und welche Dateieindung die komprimierte Datei hat.

[RLedException.py](#)

Enthält den „Fehler“ welcher geworfen wird, wenn bei der Komprimierung oder Dekomprimierung ein Fehler entsteht.

[Zusammenarbeit der Klassen](#)

Die Klassen *RL2* und *RL3* erben von der Klasse *RL* aus der Datei *RLedInterface.py*. Der *RLedError* aus der Datei *RLedException.py* wird von *RL2* und *RL3*, sowie der *start.py*, geworfen. Die *RLedSchnittstelle.py* greift auf die Klassen *RL2* und *RL3* zu und erstellt ein Objekt, um die jeweilige Aktion auszuführen. Durch die *start.py* werden die Konsolenparameter verarbeitet und die jeweilige Aktion in der *RLedSchnittstelle.py* ausgeführt.

[Programmtests](#)

Das Komprimieren und Dekomprimieren war mit allen uns zur Verfügung gestellten Dateien erfolgreich (sofern möglich).

[Erkenntnisgewinn](#)

Es ist anstrengend mit Dateien zu arbeiten, welche in Bytes verarbeitet werden müssen. Auch ist (meiner Meinung nach) die Verwendung der *RLedSchnittstelle.py* nicht sehr sinnvoll gewesen.