***Rucksack-Problem Solver***

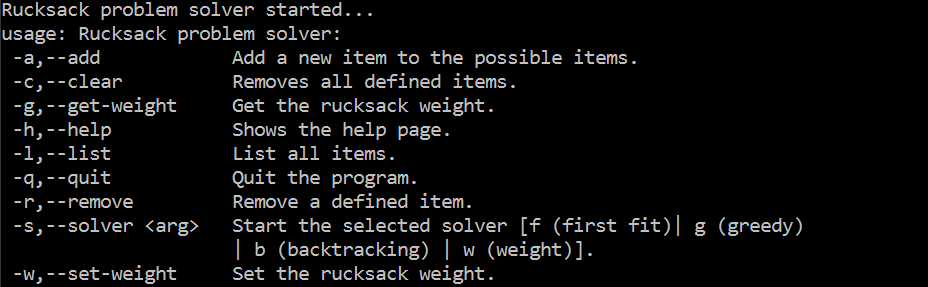
Das folgende Programm löst das Rucksackproblem unter Verwendung unterschiedlicher Algorithmen. Es werden vier verschiedene Algorithmen angeboten:

* ***First Fit:*** Dabei werden die Gegenstände der Reihe nach versucht in den Rucksack zu packen, jeder Gegenstand der hineinpasst, wird aufgenommen. Dabei wird die Wertigkeit der verschiedenen Gegenstände nicht berücksichtigt und das Ergebnis ist meist nicht sehr gut.
* ***Greedy:*** Bei diesem Algorithmus werden die Gegenstände zunächst der Wertigkeit nach sortiert (hohe Wertigkeit zuerst) und anschließend versucht jeweils der mit der größtmöglichen Wertigkeit in den Rucksack zu packen. Passt ein Gegenstand nicht, wird als nächstes der Gegenstand mit der nächstkleineren Wertigkeit ausprobiert. Hierbei wird wenig Wert auf das Gewicht der unterschiedlichen Gegenstände gelegt.
* ***Weight:*** Zunächst werden alle Gegenstände dem Gewicht nach sortiert (kleinere zuerst) und anschließend probiert, zuerst die kleinen Gegenstände in den Rucksack zu packen. Dadurch wird versucht so viel Gegenstände wie möglich in den Rucksack zu packen und über die Anzahl eine möglichst hohe Wertigkeit zu erzielen.
* ***Backtracking:*** Beim Backtracking werden alle unterschiedlichen Packmöglichkeiten für den Rucksack angeschaut und die Bestmögliche ausgegeben.

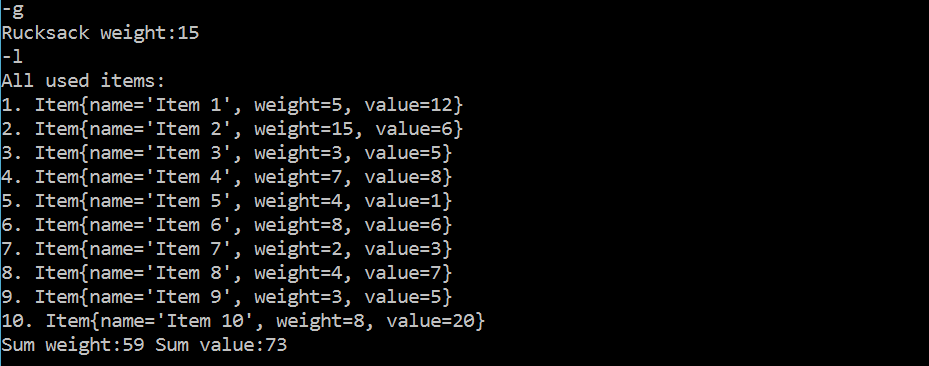
Die Anwendung wurde in **Java** implementiert und benötigt daher eine **Java Runtime**. Das Benutzerinterface wurde mittels Konsolenausgabe (Apache Commons CLI) implementiert. Um die Anwendung zu starten muss die **jar** Datei in die Runtime Umgebung geladen werden, hierzu muss das folgende Kommando in der Konsole (gilt für Windows 10) ausgeführt werden:

java -jar RucksackProblem.jar

Als erstes wird dann eine Hilfe mit den unterschiedlichen Funktionen der Anwendung dargestellt.

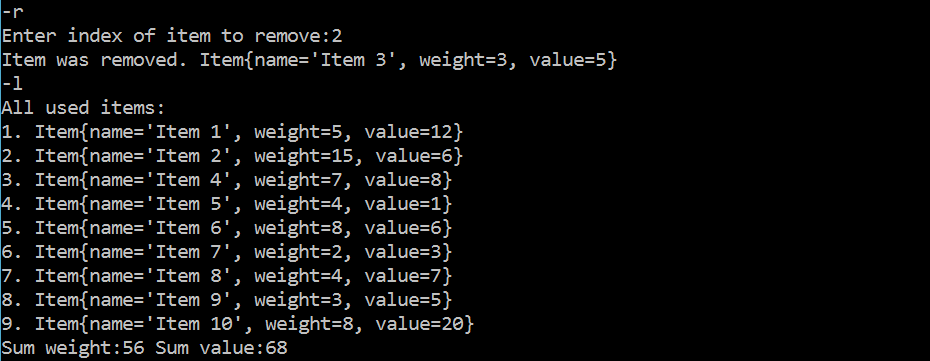


Um eine möglichst einfache Bedienung von Anfang an zu gewährleisten wurde die Rucksackkapazität bereits auf **15** gestellt und **10** unterschiedliche Gegenstände eingefügt. Die beiden Eingaben können mit *-g* (get-weight) und *-l* (list items) abgefragt werden.

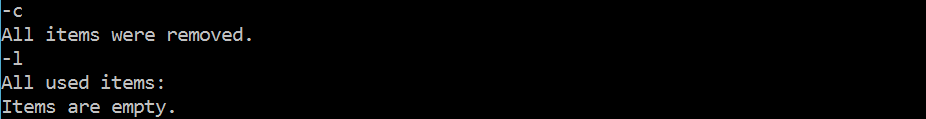


Aufgrund des Gesamtgewichts (59) aller Gegenstände ist ersichtlich, dass nicht alle Gegenstände in den Rucksack gelegt werden können, da dieser eine geringere Kapazität aufweist (15).

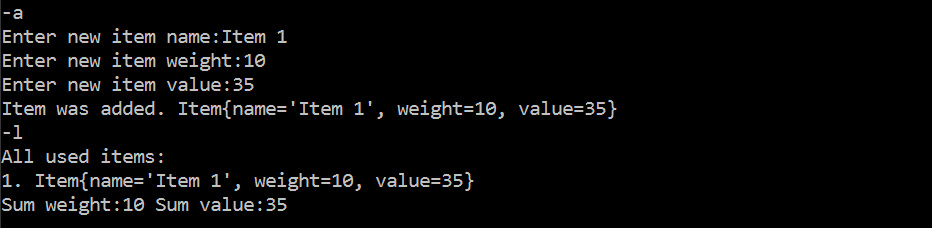
Sollten die bestehenden Gegenstände nicht passen, kann mittels *-r* (remove) Befehl ein einzelner Gegenstand aus der Liste entfernt werden.



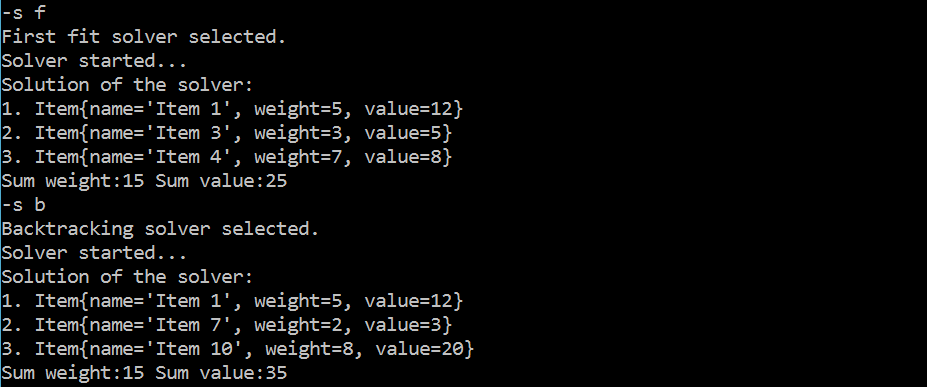
Eine vollständige Löschung der vorhandenen Gegenstände kann mit *-c* (clear) durchgeführt werden.



Ein neuer Gegenstand kann dann mittels *-a* (add) Befehl hinzugefügt werden.



Nachdem nun alle Gegenstände sowie die Rucksackkapazität den Bedürfnissen entsprechend angepasst wurden. Kann mittels *-s* (solver) Kommando und zugehörigem Solvertyp (f, g, w, b) das Rucksackbepacken gestartet werden.



In dem obigen Beispiel hat der Backtracking Algorithmus ein besseres Ergebnis als der First Fit Algorithmus geliefert.

Zudem ist es ganz einfach, die bestehende Rucksackkapazität zu verändern und die Solver erneut zu starten.

