coding contest 2016

KNAPP

# AUFGABENSTELLUNG

## ALLGEMEINE INFORMATIONEN

Die nachfolgend beschriebene Aufgabenstellung ist als Einzelarbeit zu lösen. Wie im realen Programmieralltag ist es erlaubt, das Internet zur Recherche zu verwenden. Wenn Du Fragen zur Problemstellung hast, wende Dich bitte an einen der KNAPP-Betreuer.

Die Aufgabenstellung ist innerhalb von 2,5 Stunden zu lösen. Mehrfache Abgaben sind möglich und sinnvoll, es wird die beste Abgabe für die Platzierung berücksichtigt.

## EINLEITUNG

In einem Verteilzentrum werden Kundenbestellungen, die mehrere Produkte in verschiedenen Mengen beinhalten, in einem Paket zusammengepackt und verschickt. Diesen Vorgang nennt man auch *Kommissionieren*.

Da die Kundenbestellungen rasch beim Kunden ankommen sollen, haben Verteilzentren einen Bereich in dem effizient kommissioniert wird, und einen, in dem große Mengen an Waren als Vorrat gelagert werden. Um die Produkte kommissionieren zu können, müssen diese aus dem Vorratsbereich in den Kommissionierbereich transportiert werden. Diesen Vorgang nennt man *Nachschub* oder *nachschieben*.

Im Vorratsbereich werden die Produkte in großen Mengen auf Paletten gelagert. Diese Paletten befinden sich meist in Hochregalen, man benötigt also einen Stapler, um diese zu erreichen. Die Lagerorte im Vorratsbereich sind für dieses Beispiel nicht von Bedeutung und auch nicht abgebildet. Von den Produkten ist für dieses Beispiel im Vorratsbereich eine unbeschränkte Menge vorhanden.

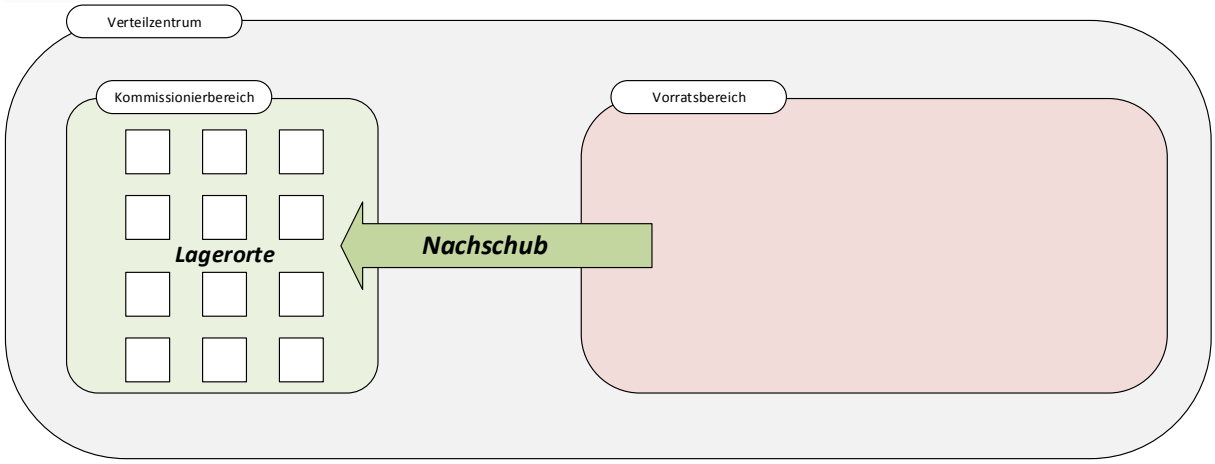


**Abbildung 1 - Schema Verteilzentrum**

Im Kommissionierbereich werden die Waren für die Kundenaufträge zusammengestellt. Das soll so schnell wie möglich erfolgen, deshalb sind alle Lagerorte in diesem Bereich ohne weitere technische Hilfsmittel erreichbar. Dies bedingt aber auch, dass nur eine kleinere Menge an Waren in diesen Lagerorten gelagert werden kann. Die maximale Menge ist aufgrund der unterschiedlichen Größen der Produkte unterschiedlich und ist pro Produkt vorgegeben.

Daher müssen die Lagerorte im Kommissionierbereich regelmäßig mit den Waren, die für die Kundenaufträge benötigt werden, nachgefüllt werden.

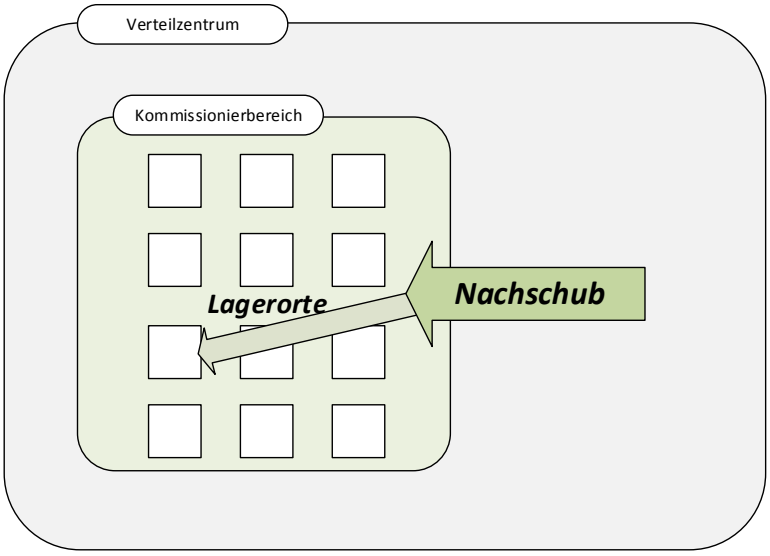


**Abbildung 2 - Nachschub für mehrere Lagerorte**

Dies geschieht mit Nachschubaufträgen, mit denen Produkte aus dem Vorratsbereich in den Kommissionierbereich gebracht werden und die Lagerorte aufgefüllt werden.

Für diesen Vorgang sind eigene Mitarbeiter eingeteilt, die sich nur darum kümmern. Erst wenn alle Produkte die für eine Kundenbestellung benötigt werden im Kommissionierbereich verfügbar sind, kann die Kundenbestellung bearbeitet werden.

## AUFGABE

Deine Aufgabe ist es, ein Software-Modul zu entwickeln, das den Nachschub steuert und somit die Kommissionierung von Aufträgen ermöglicht. Dabei

- darf maximal ein Produkt mit mehreren Stück pro Nachschub nachgelagert werden,
- darf maximal ein Produkt an einem Lagerort gelagert sein,
- darf die maximale Anzahl eines Produktes an einem Lagerort nicht überschritten werden,
- sollen die Kundenbestellungen schnell fertig gepackt sein.

# ERLÄUTERUNGEN

## Produkt

Produkte sind Waren, die gehandelt werden und voneinander unterscheidbar sind. Ein rotes Hemd, Größe 42 ist ein Produkt, es unterscheidet sich von einem roten Hemd Größe 43.

Ein Produkt hat einen Code, der es eindeutig identifiziert.

## Lagerort

An einem Lagerort wird ein Produkt in einer bestimmten Menge gelagert. Die Menge wird durch Nachschub erhöht, durch Kommissionierung verringert. Wenn am Lagerort kein Stück eines Produktes vorhanden ist, kann der Lagerort für ein anderes Produkt verwendet werden. Es dürfen nicht verschiedene Produkte gleichzeitig an einem Lagerort gelagert werden.

Für diese Aufgabe gibt es – außer dem Code zur Identifikation – keine Unterschiede zwischen den Lagerorten.

## Nachschubauftrag

Ein Nachschubauftrag wird durch dazu eingeteilte Mitarbeiter ausgeführt. In diesem Beispiellager kann pro Zyklus ein Nachschubauftrag ausgeführt werden, der durch „Dein Softwaremodul“ erstellt wurde.

Ein Nachschubauftrag kann nur ein Produkt beinhalten. Im Nachschubauftrag müssen folgende Informationen für jedes zu bewegende Produkt enthalten sein:

- Produktcode – welches Produkt nachgefüllt werden soll
- Lagerort – in den das Produkt nachgefüllt werden soll
- Menge – Anzahl an Stück des Produktes, die nachgefüllt werden sollen

Die Auftragsnummer wird vom KNAPP-Code erzeugt und hat keine weitere Bedeutung.

## ABLAUF

Die anfangs leeren Lagerorte im Kommissionierbereich werden nach und nach aufgefüllt.[1]

Dazu wird vom KNAPP-Code die Methode `GetNextReplenishmentOrder()` aufgerufen. In dieser Methode soll der von Dir implementierte Code ein Produkt, den Lagerort und die Menge errechnen, die nachgeschoben werden soll. Bei Produkt und Menge solltest Du überlegen, was in welcher Menge benötigt wird, und wie viele Stück maximal am Lagerort gelagert werden können.[2] Bei der Auswahl des Lagerortes musst Du darauf achten, dass der Lagerort auch frei ist. Es ist auch möglich in einem Zyklus keinen Nachschub durchzuführen.

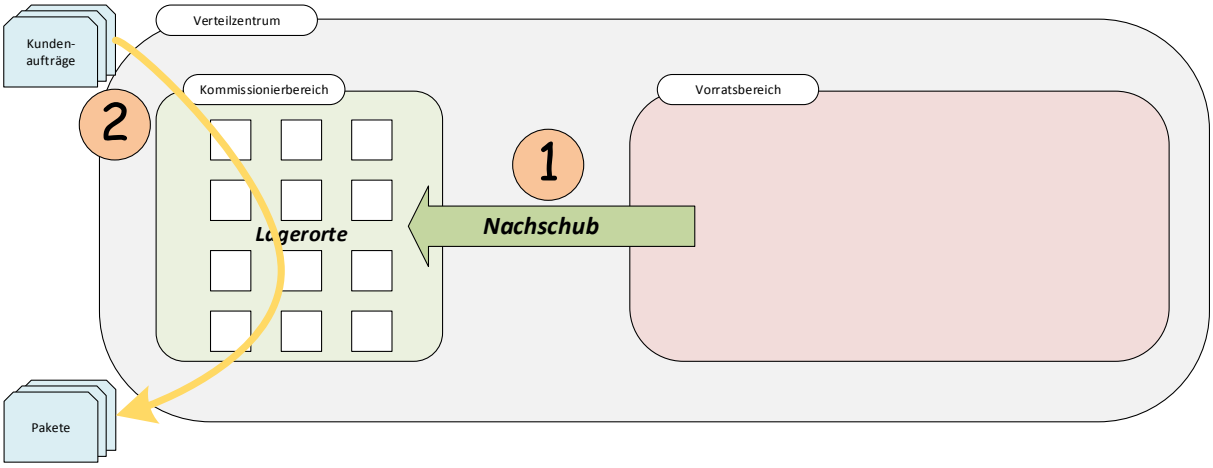


**Abbildung 3 - Ablauf**

Nachdem der Nachschub durchgeführt wurde (1), wird ein Kundenauftrag, der mit den Produkten, die sich im Kommissionierbereich befinden, erfüllt werden kann, abgearbeitet und versandt (2). Wenn für keinen Kundenauftrag alle Artikel verfügbar sind, wird in diesem Zyklus kein Kundenauftrag abgearbeitet.

Die Kundenaufträge sind von Anfang an bekannt, es kommen über die Laufzeit keine Kundenaufträge hinzu.

[1] Jeweils ein Aufruf der Funktion `GetNextReplenishmentOrder`
[2] Diese Menge ist vom Produkt selbst abhängig und ist in dessen Eigenschaft `MaxLocationQuantity` hinterlegt

Das Zubuchen der Mengen in die Lagerorte nach erfolgtem Nachschub, sowie die Auswahl der Kommissionieraufträge und das Kommissionieren erfolgen im bereitgestellten Code und muss nicht von Dir implementiert werden.

## KOMMISSIONIERUNG

Die Kommissionierung erfolgt durch den KNAPP-Code nach dem Nachschub.

Es werden die Kundenaufträge der Reihe nach durchsucht und der erste Auftrag, bei dem alle Zeilen bedient werden können, kommissioniert. Dabei werden immer alle Lagerorte eines Produktes zusammengezählt.

Wenn ein Produkt auf mehreren Lagerorten vorhanden ist, so wird zuerst der Lagerort verwendet, auf dem weniger Stück vorhanden sind.

Bei der Kommissionierung wird die Stückzahl eines Produktes, die für den Kundenauftrag benötigt wird aus dem Lagerort entnommen und von der Stückzahl am Lagerort abgezogen.

Es wird pro Zyklus maximal ein Kommissionierauftrag abgearbeitet.

## BEWERTUNGSSCHEMA

- Es werden nur auf den Bewertungsserver hochgeladene Ergebnisse gewertet.
- Die Resultate werden am Server mit den dort hinterlegten Algorithmen für die Kommissionierung errechnet.
    - Es werden nur Abgaben gewertet, bei denen die maximale Anzahl an Produkten und Stück am Lagerort eingehalten wurden.
- Lösungen mit einem früheren Ende der Kommissionierung sind besser.
    - Die Kommissionierung ist dann beendet, wenn alle Kommissionieraufträge bearbeitet werden konnten.
- Lösungen die früher abgegeben werden sind besser.

Bei mehreren Abgaben (Uploads) zählt immer die beste Abgabe zu dem Zeitpunkt an dem diese getätigt wurde. Wenn Du nach dem Upload einer Lösung weiter arbeitest und durch Deine Änderungen das Ergebnis schlechter wird, bleibt die zweite Abgabe unberücksichtigt.

## ABGABEMODUS

Zur Beurteilung des Ergebnisses wird das Ergebnis-CSV-File jedes Teilnehmers von KNAPP geparst. Dazu muss ein Archiv, das zumindest die Dateien `replenishmentOrders.csv` und `KCC2016.properties` enthält, über die Abgabeseite hochgeladen werden.

Es wird vom KNAPP Code automatisch eine Datei `upload2016.zip` erstellt, die diese beiden Dateien beinhält. Du brauchst also die Dateien nicht selbst zippen und nur diese Datei hochladen.

Mehrfachabgaben sind möglich. Für das Ranking wird die jeweils beste Lösung eines Teilnehmers herangezogen.

Sofern KNAPP dies verlangt, ist der Source-Code der Lösung ebenso über die Abgabeseite hochzuladen. Sollte dies nicht erfolgen, so behält sich KNAPP vor, die Teilnehmerin oder den Teilnehmer zu disqualifizieren.

## UPLOAD WEBSITE

Unmittelbar nach dem Upload erhält jeder Teilnehmer ein detailliertes Feedback zu seiner Abgabe.

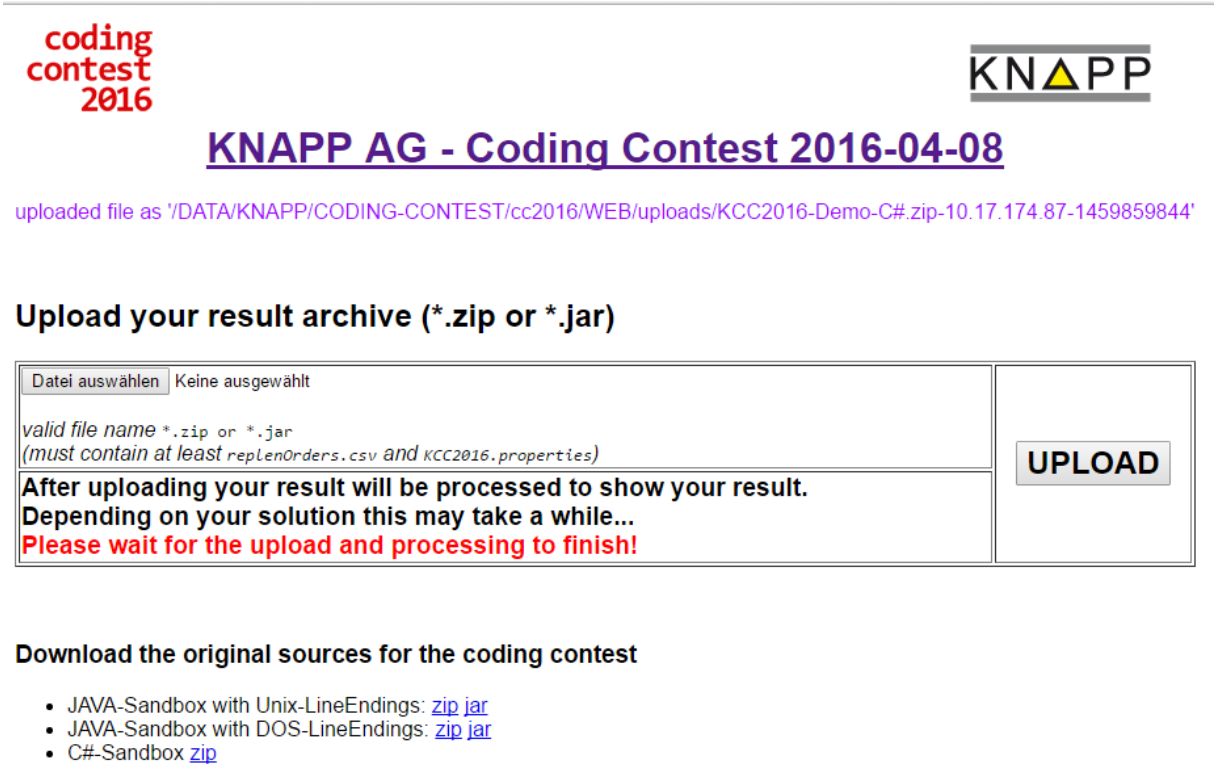


**Abbildung 4: Upload-Server (Beispiel)**

Im Feedback siehst Du die Probleme, die noch in deiner Abgabe vorhanden sind. Erst wenn alle Muss-Kriterien erfüllt sind, siehst Du auch den Score, den deine Lösung erzielt. Je höher dieser Wert ist, desto besser ist das Ergebnis.

## TIPPS

- Versuche mit Deiner Software zuerst die zwingenden Anforderungen zu lösen, erst danach solltest Du Dich dem Finden einer besseren Lösung zuwenden.
- Schau Dir den von uns vorgegeben Code an und prüfe, ob Du Teile wiederverwenden oder erweitern kannst.
- Du kannst alle Teile des Source-Codes verändern. Die Abgabedatei muss jedoch dem vorgegebenen Format entsprechen.
- Lade öfters hoch – es wird nur die beste Abgabe bewertet.
- Eine mögliche Vorgehensweise bei der Lösung wäre (dies ist ein Vorschlag, es wird nicht geprüft, ob Du in dieser Reihenfolge vorgehst):
    - Rechne für alle Produkte die nötigen Mengen aus und
    - schiebe diese in den möglichen Mengen nach.

## CODE-DETAILS

Das bestehende Framework, in dem Du Deine Lösung umsetzt, liest alle nötigen Daten ein (Klasse *Input*) und stellt diese als Objekte zur Verfügung. Ebenso wird das Ergebnis automatisiert in die Lösungsdatei *replenishmentOrders.csv* geschrieben. Die Lösung und die Datei mit den persönlichen Informationen (`KCC2016.properties`) werden gemeinsam in der Datei `upload2016.zip` archiviert. Diese Datei musst Du dann auf den Abgabeserver hochladen.

Deine Lösung sollte in *`Solution.GetNextReplenishmentOrder(…)`* implementiert werden, oder zumindest von dort ihren Ausgangspunkt nehmen. Es steht dir frei, den von KNAPP bereitgestellten Code zu ändern. Beachte jedoch, dass Änderungen am Kommissionier-Algorithmus ohne Auswirkungen auf das Ergebnis bleiben, da zur Bewertung die Berechnung am Server erfolgt.

Wenn Du einen passenden Nachschubauftrag erstellt hast, gib diesen als Rückgabewert der Methode zurück. Wenn Du keinen Nachschubauftrag durchführen willst oder kannst, gib null als Rückgabewert zurück.

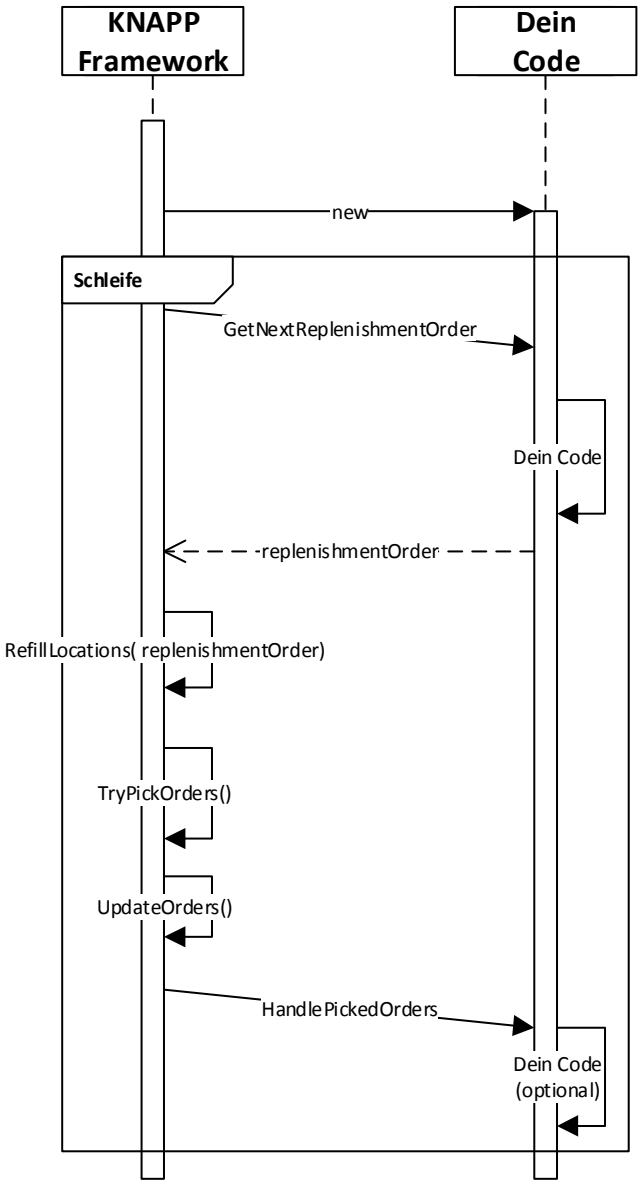


**Abbildung 5 - Ablaufdiagramm Hauptschleife**

Das Nachfüllen in die Lagerorte selbst und das bearbeiten der Kundenaufträge erfolgt im KNAPP-Code – Du brauchst dich nicht darum kümmern.

Bevor die Methode `GetNextReplenishmentOrder()` wieder aufgerufen wird, sind

- die Produkte in den Lagerorten vorhanden die nachgeschoben wurden,
- die Produkte abgezogen die kommissioniert wurden und
- die erfüllten Kundenaufträge aus der Collection entfernt.

Folgende Daten stehen Dir von Beginn an zur Verfügung:

1. alle Produkte (*products*)
2. alle Lagerorte (*locations*)
3. alle Kundenaufträge (*pick orders*) mit deren Zeilen (Produktcode und Menge)

Diese Daten werden durch den KNAPP-Code bereits eingelesen und stehen Dir in den jeweiligen Collections zur Verfügung.

Bei den Kundenaufträgen stehen Dir die Aufträge mit Zeilen, sowie eine Liste aller Zeilen aller Aufträge zur Verfügung. Der Inhalt ist ident, nach dem Kommissionieren sind sowohl die Kundenaufträge als auch die Sicht über die Zeilen wieder aktuell.


**Abbildung 6: Klassendiagramm**

..:: PLATZ FÜR NOTIZEN ::..