

Hausarbeit (Gruppen mit 3 oder 4 Personen)

Im Rahmen dieser Hausarbeit sollen die im Praktikum behandelten Themen „Grafische Programmierung von Oberflächen mit AWT und Swing“ sowie „Design Pattern“ vertieft werden.

Überblick über die Aufgabenstellung

Es soll eine einfache Lösung zur Distributions-, bzw. Lagerverwaltung implementiert werden. Im Detail soll die Anwendung die folgenden Funktionalitäten unterstützen:

- Es sollen mehrere Materiallager verwaltet werden können. Diese haben je einen Namen, **der vom Benutzer geändert werden kann**. Weiterhin sind sie in einer geografischen Hierarchie einsortiert, so dass unterhalb eines Lagerorts weitere Lager angelegt werden können. Diese Hierarchie soll grundsätzlich beliebig sein. Allein die Lager auf der untersten Hierarchieebene sollen in der Lage sein, einen aktuellen Bestand (eine Menge) zu verwalten. Eine initiale Lagerhierarchie soll zu Beginn des Programmlaufs **mit einer Default-Konfiguration (siehe Beispielscreenshots)** eingerichtet werden.

Beispiel:

```
Lager A
  Lager A.1 (mit Bestand x, Kapazität a)
  Lager A.2
    Lager A.2.1 (mit Bestand y, Kapazität b)
    Lager A.2.2 (mit Bestand z, Kapazität c)
Lager B
  Lager B.1
    Lager B.1.1 (mit Bestand w, Kapazität d)
```

- Die aktuelle hierarchische Struktur der Lager soll angezeigt werden können.
- Ein Lager enthält dabei unabhängig von der Hierarchie im Bestandssaldo immer den eigenen, bzw. die aggregierten Bestände aller Lager unterhalb dieses Lagers.
- Ein Lager verfügt über eine Maximalkapazität, die nicht überschritten werden kann.
- Ermöglichen Sie das Verteilen einer *Zulieferung* von Gütern in Form von einer Anzahl *Buchungen* auf verschiedene Lager. Dabei soll bei einer Zubuchung immer die Verteilung der Gesamtmenge auf beliebige Lager (solche, die einen Bestand verwalten können) auf beliebigen Hierarchieebenen möglich sein. Die Verteilung soll unter Angabe von Prozentsätzen der Gesamtmenge einer Lieferung erfolgen. Es kann nur eine ganzzahlige Menge Einheiten verteilt werden, mindestens jedoch 1 Einheit. Beispiel für eine Zubuchung: 1000 Einheiten werden zu 50% auf *Lager A.1*, zu 25% auf *Lager A.2.1* und 25% auf *Lager B.1.1* verteilt. Nutzen Sie zur Auswahl der Prozentsätze eine geeignete GUI-Repräsentation. Beachten Sie dabei, dass die Schrittweite prozentualer Angaben von der Anzahl der verteilten Einheiten abhängt. Beispielsweise beträgt sie bei 5 Einheiten 20%.
- Es muss sichergestellt werden, dass die gesamte Zulieferung immer vollständig verteilt wird.
- Es können keine Zulieferungen erstellt werden, die größer sind als die maximal verbleibende Lagerkapazität aller Lager zusammen.
- Implementieren Sie einen Undo-/Redo-Mechanismus (ähnlich z. B. zu Word). Dieser soll ermöglichen, dass beim Verteilen einer Zulieferung beliebig viele einzelne Buchungen schrittweise rückgängig gemacht, bzw. wieder ausgeführt werden können. Ist die Verteilung der Lieferung abgeschlossen, können keine Einzelschritte mehr rückgängig gemacht werden.
- Ermöglichen Sie *Auslieferungen* vom Bestand. Dabei wird eine Lieferung aus Abbuchungen der Bestände beliebiger Lager zusammengestellt.
- Zu- und Auslieferungen werden über das Lieferdatum eindeutig identifiziert.
- Ermöglichen Sie die folgenden Ansichten.
 - Bei Auswahl eines Lagers in der Hierarchie soll der Bestands-Saldo sowie alle Zu- oder Abbuchungen zu diesem Lager angezeigt werden, so dass ersichtlich wird, welche Buchungen direkt auf dieses Lager einwirken. Gibt es nachgeordnete Lager, soll stattdessen die Summe der Bestände dieser Lager angezeigt werden können.
 - Alle Lieferungen sollen als Liste dargestellt werden können. Bei Auswahl einer Lieferung sollen die Buchungsdetails angezeigt werden.

Gruppen mit 4 Mitgliedern bearbeiten zusätzlich die folgenden Anforderungen:

- Zu einem Lager sollen weitere Lager unterhalb des Lagers hinzugefügt werden können. Befindet sich das erweiterte Lager auf der untersten Hierarchieebene, wird der Bestand des Lagers automatisch auf das neu erzeugte Lager verschoben. Überlegen Sie, wie Sie mit der Löschung von Lagern der untersten Hierarchieebene und mit der Löschung von Lagern aus höheren Hierarchieebenen umgehen. Lager aus der obersten Hierarchieebene, die keine Lager unter sich haben können nur dann gelöscht werden, wenn Sie keinen Bestand mehr haben
- Der aktuelle Systemzustand soll in einer Datei gespeichert werden können. Dateien mit gespeicherten Systemzuständen sollen im Dateisystem gesucht und geladen werden können.

Führen Sie die folgenden Aufgaben durch:

- a) Erstellen Sie Skizzen als Prototypen für die einzelnen Oberflächen. Diese können Sie gern handschriftlich erstellen und dann einscannen.
- b) Erstellen Sie ein Klassenmodell für die fachliche Aufgabenstellung (ohne GUI-Klassen). Bei Ihrer Implementierung sollten Sie neben dem MVC Pattern mindestens zwei weitere Pattern anwenden. Nennen Sie diese Pattern und beschreiben Sie den konkreten Einsatzzweck sowie den Vorteil beim Gebrauch des Patterns in Ihrem Code. Erläutern Sie dabei auch, wie Sie das MVC Pattern angewendet haben.
- c) Implementieren Sie die Aufgabenstellung in Java unter Verwendung von AWT oder Swing. Achten Sie auf angemessene Benutzerfreundlichkeit und Robustheit.
- d) Führen Sie eine ausreichende und angemessene Kommentierung des Quellcodes und Kommentierung der Erläuterung der Klassen und Methoden anhand von Javadoc durch.
- e) Implementieren Sie das Beispiel auf der zweiten Seite.

Abgabe der Hausarbeit:

- a) Bearbeiten Sie die Aufgabe in 3er-Gruppen oder 4er-Gruppen. 4er-Gruppen bearbeiten die oben genannten Zusatzaufgaben.
- b) Abgabetermin ist Mittwoch, der 23.12.2015, 23:59 Uhr, in der entsprechenden ILIAS-Übung. Bitte legen Sie die Eigenständigkeitserklärung als PDF bei.
- c) Damit ggf. eine individuelle Benotung möglich ist, müssen sowohl Dokumentation wie auch Programmierung von allen Teampartnern in ähnlichen Anteilen bearbeitet werden. Die Projektdokumentation soll von jedem Teilnehmer einzeln angefertigt werden und das Gruppenprojekt Arbeit mit Fokus auf dem eigenen Anteil beschreiben.
- d) Die Hausarbeit wird ausschließlich in folgenden Formaten akzeptiert:
 - a. Code: Quellcode-Export von Eclipse" als ZIP-Datei, lauffähig mit JRE 1.7.x
 - b. Dokumentation: Eine einzelne PDF-Datei, durchgängig im Format A4 hochkant.
- e) Der reguläre Umfang der individuellen Textabgabe beträgt 2 Seiten. Bitte klären Sie Abweichungen mit dem Dozenten ab.

Beispiele Hierarchie und Buchungen:

Lagersystematik:

Lager	Name
1.	Deutschland
1.1	Niedersachsen
1.1.1	Hannover-Misburg
1.1.2	Nienburg
1.2	NRW
1.3	Bremen
1.4	Hessen
1.5	Sachsen
1.6	Brandenburg
1.7	MV
2.	Europa
2.1	Frankreich
2.1.1	Paris-Nord
2.1.2	Orléans

2.1.3	Marseille
2.1.4	Nîmes
2.2	Italien
2.2.1	Mailand
2.2.2	L'Aquila
2.3	Spanien
3.	Großbritannien

Buchungen:

Nr.	Menge	An- teil 1	Lager 1	An- teil 2	Lager 2	An- teil 3	Lager 3	An- teil 4	Lager 4	An- teil 5	Lager 5	Summe Anteile
1	1.000	50%	Bremen	20%	MV	10%	Mailand	10%	Spanien	10%	Großbri- tannien	100%
2	2.000	50%	Nienburg	20%	NRW	20%	Hessen	10%	Sachsen			100%
3	10.000	20%	Branden- burg	10%	Orléans	25%	L'Aquila	25%	Spanien	20%	Großbri- tannien	100%
4	5.000	50%	Nîmes	40%	MV	10%	Nienburg					100%
5	12.500	30%	Paris-Nord	20%	Branden- burg	15%	Hannover- Misburg	15%	Bremen	20%	Mailand	100%