



B2.1 Angewandte Programmierung (SL) - 1. Zug- SoSe2018

Dashboard ► ANGEW PROG-137471-9 ► 9. April - 15. April ► Ub1 Basisklassen und Generics

Ub1 Basisklassen und Generics

Übung 2: Basisklassen und Generics

12 Punkte

Lernziele:

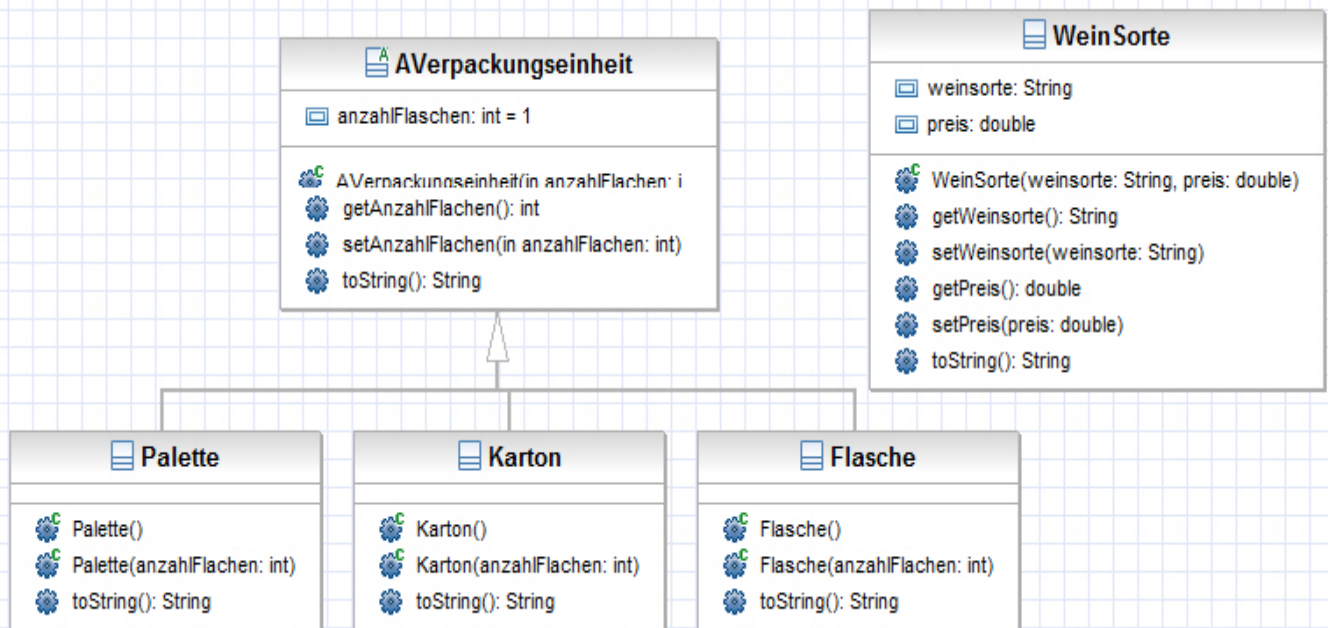
1. Vererbung
2. Arbeiten mit **generischen Klassen**, Verwendung von **generischen Collection-Klassen**
3. Arbeiten mit **abstrakten Klassen**
4. Arbeiten mit **Interfaces**
5. Pattern "Singleton"
6. **"main" schmal halten!**
7. **Codewiederholung vermeiden.**

1.Aufgabe (2 Pkte): Paket "model"

Für einen Weinhändler ist eine Software zu entwickeln, die den Vertrieb der Weine unterstützt. Die Lieferung der Weine erfolgt in unterschiedlich großen Verpackungseinheiten (Einzelflaschen, Kartons und Paletten).

Es gelten folgende Defaultwerte bei den **Verpackungseinheiten**:

1 Karton =12 Flaschen
1 Palette =36 Flaschen

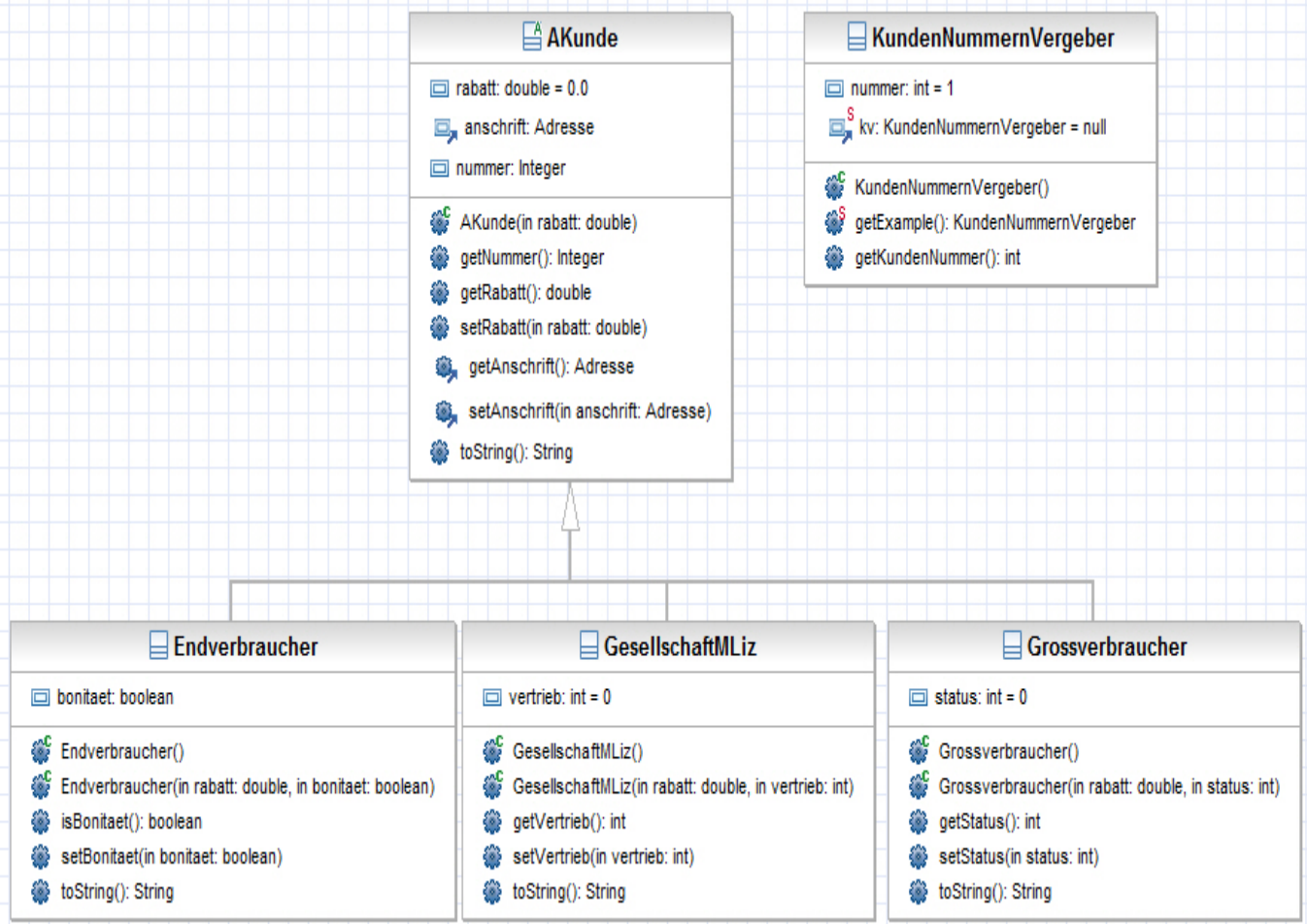


Hinweis: Das UML-Diagramm dient der Orientierung! Die dargestellten Klassen können weitere Attribute und Methoden erhalten.

2.Aufgabe (4 Pkte): Paket "kunde"

Der Weinhändler liefert an verschiedene Typen von Kunden (End- / Großverbraucher und Gesellschaften mit Lizenzvertrieb), denen unterschiedliche Rabatte gewährt werden. Es gelten folgende Defaultwerte bei der **Rabattstaffel**:

Rabatt Endverbraucher	0%
Rabatt Grossverbraucher	5%
Rabatt Lizenzvertrieb	7%



Hinweis: Das UML-Diagramm dient der Orientierung! Den dargestellten Klassen können Sie weitere Attribute und Methoden hinzufügen.

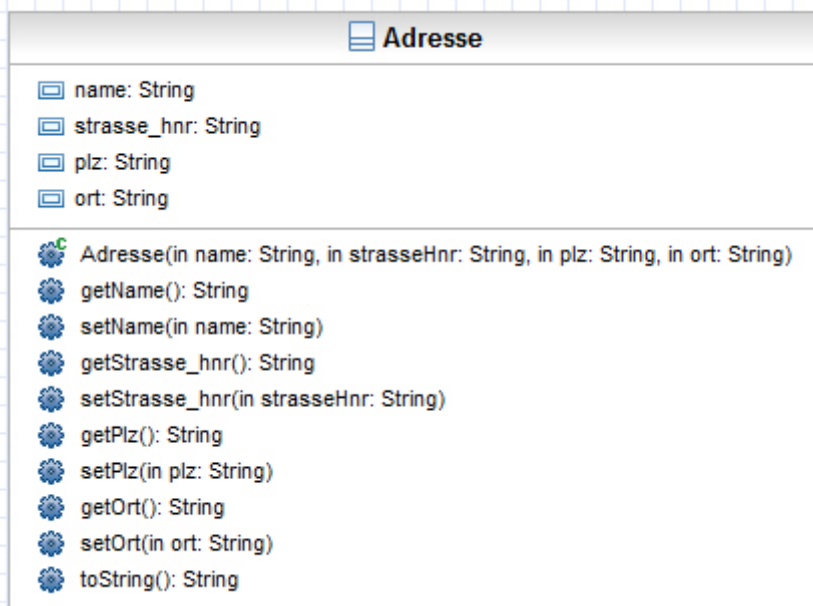
ACHTUNG: Die **Kundennummer** soll den Datentyp **String** erhalten (nicht Integer) !

Die **Klasse KundenNummernVergeber** soll als **Singleton (Entwurfsmuster)** implementiert werden! Machen Sie deshalb den **Konstruktor privat**, so dass verhindert wird, dass von außen eine Instanz dieser Klasse angelegt werden kann. Eine Instanz der Klasse wird über die Methode `getExemplar()` bereit gestellt.

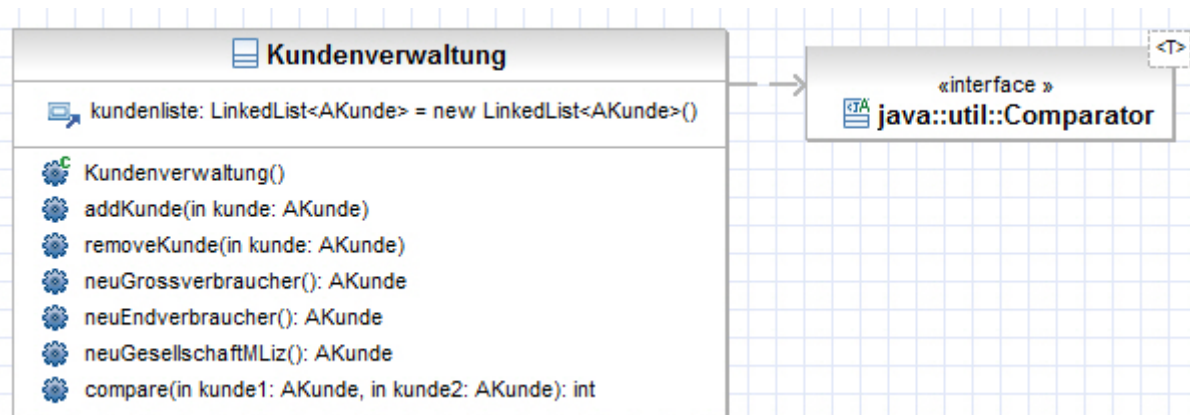
Diese Klassen sorgt dafür, dass jede **Kundennummer eindeutig** ist. Die Kundennummern werden **automatisch vergeben**.

Die **Kundennummer (String)** wird aus **Groß/Kleinbuchstaben und Ziffern** erzeugt, die **zufällig** gewählt werden. Diese ergeben eine **neunstellige** Kundennummer zur Kundenidentifikation, z.B. Kundennummer: **"abc765KLM"** Achten Sie darauf, dass die Nummer immer mit drei Kleinbuchstaben beginnt, gefolgt von drei Ziffern, gefolgt von drei Großbuchstaben.

Jeder Kunde hat eine **Adresse**, die über ein eigenes Objekt angelegt wird. Achten Sie auf **Nullwerte!**



Alle Kunden werden in einer **LinkedList** der **Kundenverwaltung** abgespeichert. Achten Sie darauf, dass Sie den **generischen Datentyp** verwenden -bitte nicht den "raw type"!



Hinweis: Das UML-Diagramm dient der Orientierung! Die dargestellten Klassen können weitere Attribute und Methoden erhalten.

Die Kundenverwaltung implementiert das **Interface java.util.Comparator <AKunde>**, damit lassen sich Kunden vergleichen. Vergleichskriterium ist die Kundennummer. Die **public int compare(AKunde kunde1, AKunde kunde2)**-Methode gibt folgendes zurück:

- 0 = beide Kundennummern gleich
- <0 = Nummer von kunde1 kleiner als Nummer von kunde2
- >0 = Nummer von kunde1 größer als Nummer von kunde2

3.Aufgabe (6 Pkte): TEST

Schreiben Sie einen **Testrahmen**, um Kunden und Bestellungen zu testen. Ein **Menu** soll folgendes für den Benutzer zur Auswahl stellen:

1. einen neuen Endverbraucher anlegen
2. einen neuen Großverbraucher anlegen
3. eine neue Gesellschaft mit Lizenzvertrieb anlegen
4. [Test] **n** Kunden **automatisch** erzeugen (n vom Benutzer einzugeben), wobei es eine zufällige Verteilung von Kundentypen geben soll.
5. [Test] alle Kunden **nach Nummer sortiert** auf der Konsole ausgeben
6. ...hier Platz für Erweiterungen...

Hinweise:

- Bitte **keine Umlaute** und **kein ß** in **Bezeichnern** oder **Dateinamen** verwenden!
- Bitte **entfernen** Sie im Quellcode **alle** Zeilen, die **Annotationen** (beginnend mit "@", beispielsweise "@Override", "@SuppressWarnings", etc.) enthalten.
- Falls Sie es nicht schon getan haben, legen Sie **Pakete** an!
- Halten Sie die **main**-Funktion **schmal**!
- **Dokumentieren** Sie Ihre Anwendung im **JavaDoc - Style** !
- **Bei Nichtbefolgung gibt es Punktabzug!**

Zuletzt geändert: Wednesday, 25. April 2018, 16:32

Einstellungen



Kurs-Administration