

Aufgabe 1

a) Erklären Sie, was das Ziel der Anwendung ist, und wie sie arbeitet

Die Anwendung ist ein einfaches Bestellsystem für Waren von verschiedenen Händlern.

In Demo ist eine main-Methode definiert. Sie erlaubt die Eingabe von Nutzernamen und Passwort über die Commandline. Danach werden aus der Commandline Warennamen und Mengen akkumuliert bis kein Warenname mehr angegeben wird. Nun wird über die Commandline abgefragt welche Erzeugergemeinschaft für die Bestellung genutzt wird. Je nach Eingabe wird eine Klasse, die Shop extended mit den Angegebenen Anmleddaten initialisiert. Schließlich wird nur noch die Warenbestellung an den jeweiligen Shop weitergegeben.

Generell für Shops gilt:

- Es müssen folgende Funktionen definiert sein: login, check, orderOnAccount, logout
- Bei jeder Bestellung wird ein login mit den Nutzerdaten durchgeführt.
- Die implementation des Logins hängt vom Shop ab, muss aber einen boolean returnen.
- Ist der Login erfolgreich wird für jeden Artikel der bestellt wird
 - durch Aufruf von check geprüft ob der Artikel auf Lager ist.
 - durch den Aufruf von orderOnAccount bestellt, wenn er auf Lager ist.
 - auf eine watchlist gesetzt, wenn er nicht auf Lager ist.
- Danach wird logout aufgerufen.

Spezielle Implementationen:

BBG:

- login: nur eine dummy-Implementation, die immer true returned (und dies in der CL ausgibt)
- check: eine dummy-Implementation, unabhängig vom Item wird immer zurückgeben, dass die Waren auf Lager sind (und dies in der CL ausgibt)
- orderOnAccount: simuliert Netzwerlatenz durch Thread.sleep() und returned immer true
- logout: Gibt nur aus dass der Nutzer sich ausgelogged hat. Keine Logik verbaut

BHK - Unterschiede zu BBG:

- login: simuliert Netzwerlatenz vor der Aussage, dass der Login erfolgreich war und vor dem return
- check: keine Unterschiede
- orderOnAccount: simuliert keine Netzwerlatenz

- logOut: keine Unterschiede
- ~~Simulation der Netzwerklatenz: andere Char Ausgabe, kürzere Wartedauer~~

Klassen für Daten:

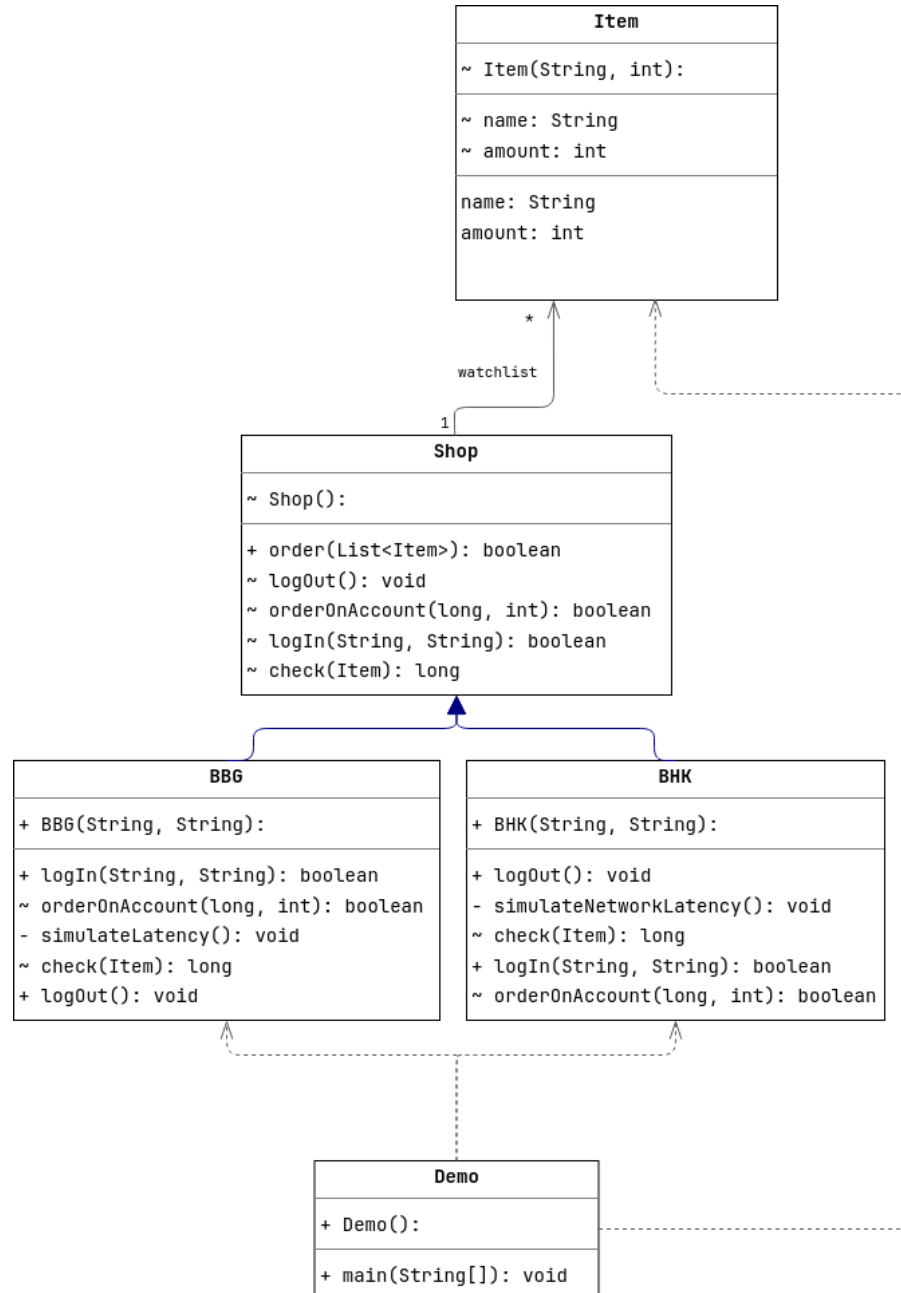
Item

- Speichert Warendaten in Form von Namen und Anzahl ab.

Zusammenfassung:

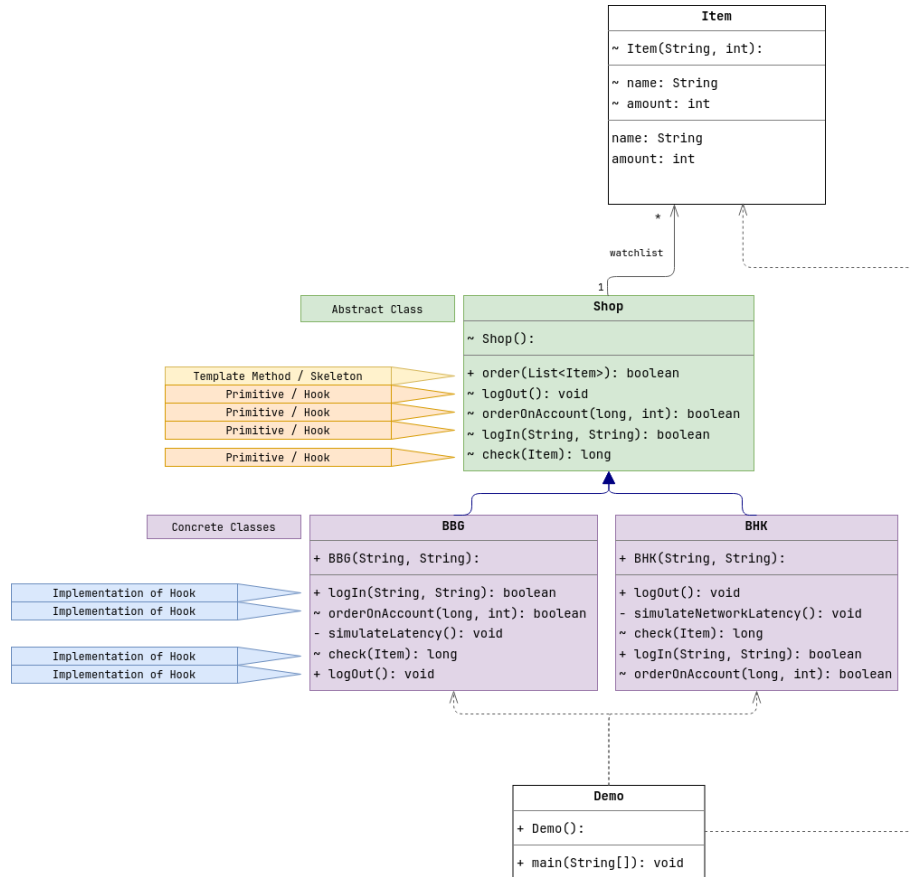
Insgesamt ermöglicht die Anwendung dem Benutzer, Waren von verschiedenen Erzeugergemeinschaften zu bestellen, wobei für jede Gemeinschaft spezifische Bestell- und Authentifizierungslogik implementiert ist.

b)



*mit Shop als «abstract»

c)



Die Implementation of Hook Annotations gelten für beide Shops

d)

Die Skeleton-Methode ist die Methode `order` in **Shop**. Sie ist in **Shop** implementiert, ruft aber Methoden auf, die noch nicht in **Shop** definiert sind - die also für jede Klasse, die von **Shop** erbt implementiert werden müssen - die "Einschub-Methoden" oder "Hooks". Die Hooks die in `order` aufgerufen werden sind `login`, `check`, `orderOnAccount` und `logout`.

Diese Vorgehensweise sorgt dafür, dass jede (erfolgreiche) Bestellung in jedem Shop immer in folgender Reihenfolge abläuft: - `login` - `check` - `orderOnAccount` - `logout`

Und jeder Shop trotzdem einen eigenen Ablauf für jede dieser Methoden implementieren kann.