

# P i z z a r a n d o

---

Paul Hufner, Sebastian Bierau, Liam Brüll

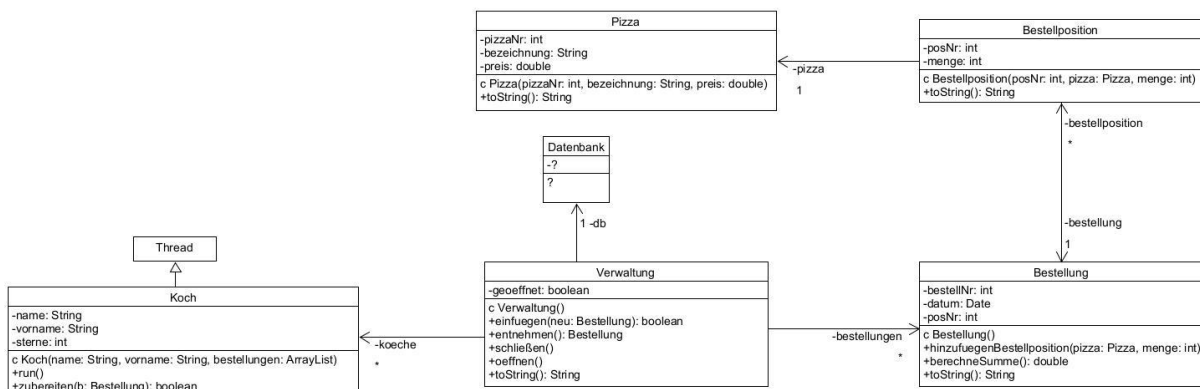
18.03.22

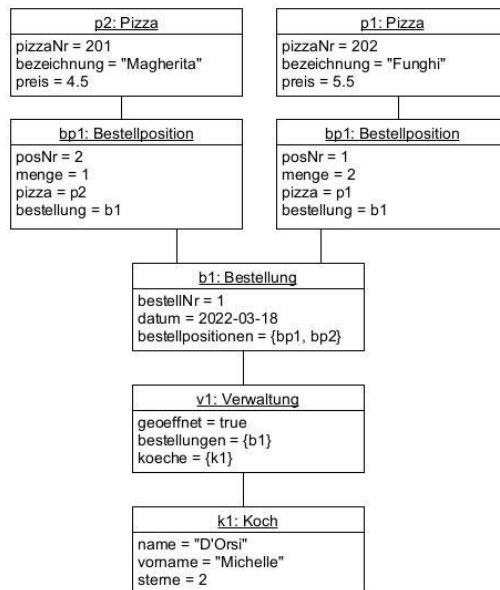
13BG PILK

Magdeburg

## Soll-Ist-Vergleich

Wir richteten uns nach der Abiturklausur B2 (Nachschreiber) von 2016: Pizzeria. Ziel war es, eine Datenbank in Verbindung mit einem Java Programm mit Client und Server zu erstellen. Zusätzlich besteht die Option eine ergänzende Website zu schreiben, die es den Benutzern ermöglicht eine Auswahl an Pizzen zu bestellen. Wir richteten uns dabei weitestgehend nach den in den Materialien der Abi Klausur bereitgestellten Klassendiagrammen, ergänzten sie jedoch um unseren Anforderungen zu entsprechen. So verwenden wir keine Queue im Java Programm, weil diese in der Datenbank leichter zu realisieren ist. Das Java Programm nimmt sich die erste Datenreihe aus der Datenbank, um immer die neueste Bestellung zu behandeln. Somit soll es nur die Klassen „Koch“, „Verwaltung“, „Bestellung“, „Bestellposition“, „Pizza“ und „Datenbank“, im Java Programm geben. Die Datenbank-Klasse soll als einzige Klasse mit der SQL-Datenbank kommunizieren und die anderen Klassen mit den jeweiligen Daten ausstatten. Eine Bestellung besteht aus beliebig vielen Bestellpositionen. Eine Bestellposition besteht aus einer Pizza-Sorte und der bestellten Menge. Bezeichnung und Menge werden im Pizzaobjekt gespeichert. Eine Bestellposition muss folglich mindestens eine Pizza beinhalten, ebenso wie eine Bestellung mindestens eine Bestellposition beinhalten muss. Die Bestellungen werden von Köchen abgearbeitet, die von der Thread-Klasse erben, damit sie zeitgleich abgearbeitet werden. Köche haben neben ihrem Namen auch Sterne, die beeinflussen, wie schnell sie Pizzen backen können. Dies wird realisiert, dadurch, dass während die Köche eine Pizza backen, der jeweilige Thread pausiert. Die Zeit, die der Thread pausiert, wird durch den Kehrwert der Sterne mal die Anzahl der Pizzen mal eine Zeitkonstante berechnet. Dies soll simulieren, dass Köche mit höherer Sterneanzahl schneller Pizzen backen können. Alle Klassen werden durch die Verwaltungsklasse verwaltet, welche die einzige Klasse ist, die mit der Datenbankklasse kommuniziert, in welcher auch die Mainmethode ist. Die Website zum Bestellen wird auf einem Java-Script Server gehostet, welcher ebenfalls auf die Datenbank zugreifen kann. Somit wird die Synchronisation der Bestellungen zwischen Server und Java Programm sichergestellt. Aktuell ist der Server in der Lage, Benutzer zu registrieren und anzumelden. Außerdem kann der Server auf die Pizzen der Datenbank zugreifen und kann sie anzeigen. Zukünftig soll er um die Funktion erweitert werden, Benutzern die Auswahl verschiedener Pizzen zu ermöglichen, welche dann in einer Bestellung der Datenbank übergeben werden. Alle weiteren genannten Anforderungen wurden mit Ausnahme der Datenbankverbindung in Java realisiert. Zum aktuellen Zeitpunkt ist das Gesamtprogramm dementsprechend nicht benutzungsfähig.





Beispielhaftes Objektdiagramm mit einer Bestellung von insgesamt 3 Pizzen, die von einem Koch bearbeitet werden.

## Beurteilung

Auch wenn wir das Ziel verfehlt haben, konnten wir viele wertvolle Erkenntnisse erarbeiten und festigen. Mit etwas mehr Arbeit könnten wir das Programm sicherlich vervollständigen, insofern wir aktuelle Probleme beheben können. Wir haben gut zusammengearbeitet und die Arbeit sinnvoll aufgeteilt. Die Datenbank soll zukünftig mit Auto-Inkrementen arbeiten, was etwas war, das wir nicht fehlerfrei implementieren konnten. Auch könnte es einen Algorithmus geben, der die Pizzen, die zubereitet werden sollen, möglichst optimal den Köchen zuteilt. Außerdem wäre eine Benutzeroberfläche für Köche praktisch, welche in der Praxis logischer als eine Wartezeit im Thread wäre.