

## **Pizzadienst**

Ihr werdet von der Pizzakette „1-2-3 Pizza-Fly“ mit der Realisierung einer Softwarelösung beauftragt. Nach den ersten Vorgesprächen habt ihr die ersten groben Anforderungen wie folgt zusammengefasst.

**Anforderungsbeschreibung:** Die Bestellung erfolgt ausschließlich telefonisch. Ein Ladengeschäft ist nicht vorgesehen. Jeder Kunde hat immer genau eine Adresse. Eine vollständige Adresse besteht immer aus einer Straße und Hausnummer, der Postleitzahl und dem Wohnort. Kunden ohne vollständige Adresse werden nicht beliefert.

Ebenso verhält es sich bei den telefonischen Kontaktdaten. Zu einem Kunden muss immer mindestens eine Telefonnummer existieren, wobei einem Kunden auch maximal zwei Telefonnummern zugeordnet sein können. Weiterhin ist zu jedem Kunden die Anrede, der Name, Vorname und die Kundennummer abzuspeichern. Jeder Mitarbeiter der Pizzakette darf auch Produkte bestellen. Eine preisliche Vergünstigung ist dafür aber nicht vorgesehen. Bei der Bestellung gelten für einen Mitarbeiter dieselben Regeln wie für einen regulären Kunden. Der Mitarbeiter besitzt, abweichend vom Kunden, eine Mitarbeiternummer.

Die Pizzakette bietet zwei unterschiedliche Produkte an. Es können Pizzen, aber auch Baguettes bestellt werden. Sowohl Baguettes als auch Pizzen haben eine unterschiedliche Anzahl von Belägen. Der Gesamtpreis eines Produktes addiert sich aus den Einzelpreisen der Beläge, sowie einem Grundpreis, der für Pizza und Baguette je nach Größe unterschiedlich sein kann. Pizzen und Baguettes haben verschiedene Größen, Pizzen können einen Durchmesser von 25, 28 und 32cm, Baguettes können halbe oder ganze Größe haben.

Zu einer Bestellung werden immer der Kunde oder Mitarbeiter, eine Bestellnummer, das Bestelldatum sowie die bestellten Produkte mit Angabe der bestellten Menge festgehalten. Zusätzlich gibt es ein Auslieferungs- und Abnahmedatum. Das Auslieferungsdatum gibt an, wann die Bestellung versendet wurde, während das Abnahmedatum den Empfang durch den Kunden belegt. Bei allen Datumsangaben sind sowohl das Datum wie auch die Uhrzeit festzuhalten.

**Aufgabe:** Entwickelt und programmiert eine objektorientierte Lösung, welche die oben beschriebenen Anforderungen exemplarisch umsetzt. Achtet auf eine möglichst robuste Implementation.

**Hinweise:** Zur Lösungen der Aufgabe ist es sinnvoll, die Aufgabe in drei Teilprobleme zu untergliedern.

Die jeweiligen Teilprobleme können dann zunächst isoliert voneinander betrachtet und gelöst werden. Liegt für jedes Teilproblem eine Lösung vor, werden diese am Ende zu einer gesamten Lösung zusammengefügt.

Als Teilprobleme können betrachtet werden:

- Kunde, Mitarbeiter, Adresse, Kontaktdaten
- Produkt, Pizza, Baguette, Beläge
- Bestellungen

Um aus der textuellen Beschreibung zu isolieren, was mögliche Beziehungen (Assoziationen) zwischen Klassen und was mögliche Vererbungsbeziehungen sind, kann im ersten Ansatz auf die sprachlichen Hilfsmittel: „hat ein/e“ und „ist ein/e“ zurückgegriffen werden. Der sprachliche Ausdruck „hat ein/e“ deutet dabei eine Assoziation an. Beispielsweise könnte man sagen: „Der Kunde hat eine Adresse.“ Dagegen deutet „ist ein/e“ eine mögliche Vererbung an, wie zum Beispiel: „Der Kunde ist eine Person.“

- Erstellt nur die absolut notwendigen `get`-Methoden.
- Achtet auf ausreichende Testabdeckung, testet alle geforderten Vor- und Nachbedingungen sowie die Invarianten.
- Dokumentiert eure Klassen und Methoden sinnvoll.