

## ▼ Aufgabe 1 - Lücken füllen

⚠ Der gesamte Code ist auskommentiert. Kommentiert ihn ein, um die Compilerfehler zu enthüllen!

Hier findet ihr eine kleine Simulation zum Pizza backen.

Löst die vielen Compilerfehler und bringt den Code zum Laufen.

**Datei für die Aufgabe:** `src > 1PizzaAutomatLückenFüllen`

## ▼ Aufgabe 2 - PizzaAutomat in Funktionen auslagern

⚠ Knobelaufgabe!

Der Pizza-Automat in Aufgabe 1 ist ein riesiger Batzen Code. Das wird schnell unübersichtlich. Deshalb wollen wir folgende Schritte in einzelne Funktionen auslagern:

1. Das Drucken der Zutatenliste in eine Funktion namens "druckeZutaten"
2. Die Erstellung des Pizzabodens bis einschließlich den Satz "Pizzateig erstellt" in der Funktion namens "pizzaTeigKneten"
3. Das Verteilen von Tomatensauce und Käse auf dem Boden in der Funktion namens "grundpizzaFertigstellen"
4. Das Eingeben der Beläge und das Ausdrucken der Belagliste in eine Funktion "belaege"
5. Den Ofencountdown in eine Funktion namens "countdown"

Kriegst du es hin? Überleg dir genau, welche Funktionen Parameter brauchen und welche nicht.

Für Tipps, öffne den Tipp-Abschnitt hier drunter.

**Datei für die Aufgabe:** `src > 2PizzaInFunktionen`

## ▼ Aufgabe 2 - Tipps

Am Ende soll die `main()` so aussehen:

```

67 ▶ fun main(){
68     var zutatenListe: List<String> = listOf("Mehl", "Hefe",
        "Wasser", "Tomatensoße", "Käse")
69
70     println("Willkommen beim Pizza-Automat.")
71
72     druckeZutaten(zutatenListe)
73     pizzaTeigKneten(zutatenListe)
74     grundPizzaFertigStellen(zutatenListe)
75     belaege()
76
77     println("Pizzavorbereitung abgeschlossen.")
78
79     ofenCountdown()
80 }

```

Die Funktionen brauchen folgende Parameter:

1. "druckeZutaten" braucht die zutatenListe
2. "grundPizza" braucht ebenfalls die zutatenListe
3. "belaege" braucht KEINE Parameter, aber muss eine List mit den eingegebenen Daten zurückgeben!
4. "ofenCountdown" braucht keine Parameter