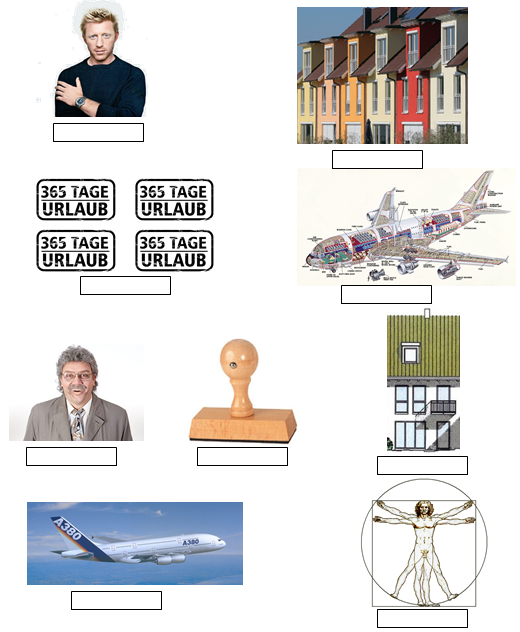
**Aufgabe zu OOP:**

**A1**

Weisen Sie den untenstehenden Bildern jeweils die Begriffe Klasse bzw. Objekt zu

[](file:///C:\schule\unterricht\web\runestone_php_oop\build\php_oop\_images\image4.png)

**Aufgabe:** Überlegen Sie sich, welche Klassen es in Ihrer beliebten Berufsschule gibt.

Finden Sie auch einige Attribute und Methoden

**A2 Fahrzeugleihe**

Arbeitsauftrag

* Erstellen Sie zwei Klassen (Auto, Mitarbeiter) zur Verwaltung eines Fuhrparks in einem Unternehmen
* Speichern Sie sich für das Auto die Attribute (Hersteller, Modell, Leistung, Verbrauch, Baujahr, KilometerStand)
* Die Klasse Auto verfügt über eine Methode fahren(). Diese prüft, ob die Türen geschlossen sind. Falls ja, wird der Motar gestartet, falls nein erhält der Nutzer einen Hinweis
* Mitarbeiter können sich Fahrzeuge für eine Dienstreise ausleihen. Bei der Rückgabe müssen Sie die gefahrenen Kilometer sowie den Kilometerstand des Autos an den Chef melden.
* Erstellen Sie zunächst ein Klassendiagramm
* Implementieren Sie anschließend den Sachverhalt mit Hilfe von PHP
* Der Mitarbeiter Brenner leiht sich für eine Dienstreise nach Hamburg (700 km einfach) den BMW mit dem Kennzeichen WÜ-MA-777 aus. Der Kilometerstand ist am Ende der Fahrt im Auto zu setzen.
* **Überlegen Sie, wie das Objekt Brenner mit dem Objekt Auto kommunizieren kann.**

**A3 HighScoreListe**

Von klassischen Computerspielen kennen wir vielleicht noch die sog. Highscore-Liste. Diese stellen die höchsten in dem entsprechenden Spiel erzielten Punkte in absteigender Form dar.

Eine Highscore-Liste füre ein Computerspiel verwaltet mehrere Einträge, die jeweils durch einen Spitznamen des Spielers repräsentiert sind. Die Highscore-Liste verfügt weiterhin über gängige Methoden, die das Hinzufügen von Platzierungen und die Ausgabe der gesamten Liste ermöglichen.

**Aufgabenstellung**

* Überlege dir ein Klassendiagramm für die beschriebenen Eigenschaften.
* Implementiere die Klasse *HighScoreEntry\**, mit der wir Einträge für die HighScore-Liste erzeugen können. Die Klasse soll eine geeignete *toString()*-Methode besitzen.
* Befülle die HighScore-Liste mit entsprechenden Daten. Eine Sortierung nach Punkten muss nur für die Ausgabe erreicht werden; die interne Implementierung bleibt Ihnen überlassen.
* Aufgrund der Verwendung von PHP sollte eine Persistierung per JSON erfolgen.
* Die Liste soll maximal 10 Einträge aufnehmen. Neue Einträge mit höheren Punktzahlen müssen dementsprechend kleine Einträge entfernen.

**A4 Kochrezeptverwaltung**

Wir wollen eine Software für eine Koch-Website entwickeln. Das Kundengespräch liefert folgende Informtionen:

“Ein Rezept hat einen Namen sowie eine Zubereitungszeit in Minuten und besteht aus einer Menge von Anweisungen. Jede Anweisung beschreibt dabei an welcher Stelle was zu tun ist. Zudem verfügt eine Rezept über eine Menge von Zutaten. Jede Zutat gibt an, was für das Rezept benötigt wird und wie viel davon benötigt wird. Für Letzteres muss zudem die jeweilige Einheit vermerkt sein. Ein Kochrezeot enthält spezische Anweisungen zum Einstellen de Herds. EIn BAckrept enthält eine spezifische Anweisung in Bezug auf das Einstellen des Backofens.”

* Erstellen Sie aus den obenstehenden Angaben ein Klassendiagramm
* Programmiere anschließend eine Testmethode, die die Einzelheiten des untenstehenden Testfalles als Objekt instanziiert und als echo()-Anweisung ausgibt.

Testfall

Pfannkuchen (30 Minuten)

Zutaten

- 80 g Kartoffelmehl

- 80 g Maisstärke

- 3 Eier

- 400 ml Milch

- 5 EL Traubenzucker

- 5 EL Pflanzenöl

Zubereitung

1. Die Mehle vermischen und sieben

2. Eier, Zucker und Milch zugeben

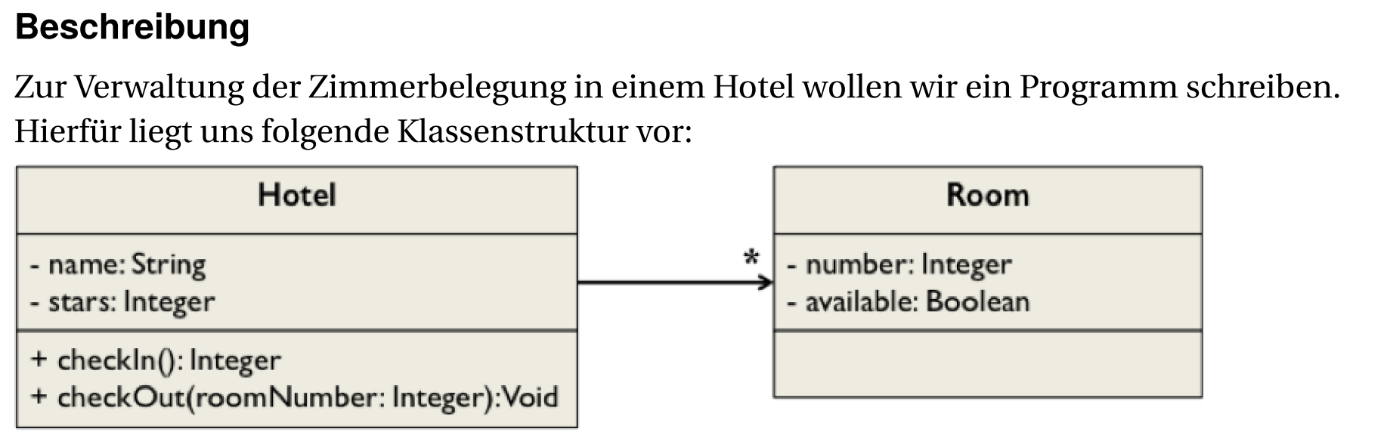
3. Alles mit dem Schneebesen gut verquirlen

4. 10 Minuten quellen lassen

5. Noch einmal verrühren

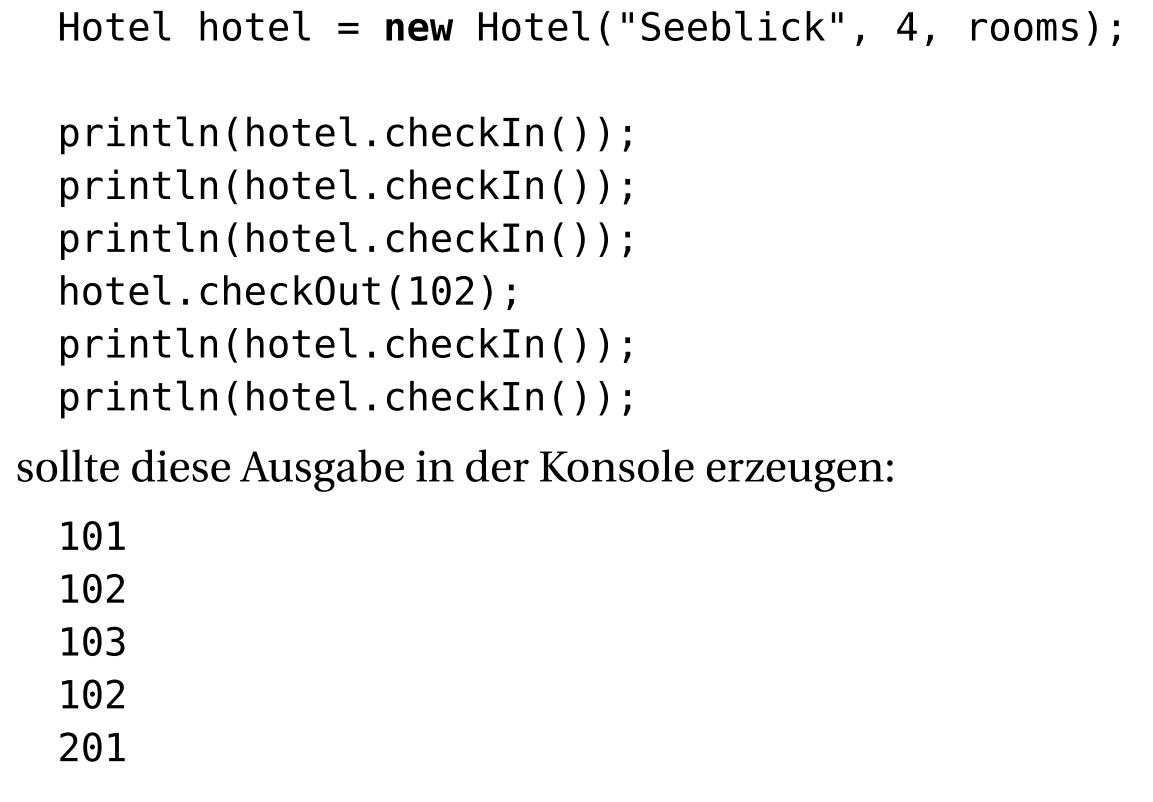
6. Pfanne auf hoher Stufe erhitzen und Teig portionsweise in heißem Öl ausbacken

**A5 Hotelzimmer**

[](file:///C:\schule\unterricht\web\runestone_php_oop\build\php_oop\_images\image14.png)

Getter- und Setter-Methoden sind nicht aufgelistet, werden aber in den entsprechenden Klassen erwartet. Für die Methoden in den *Hotel*-Klassen müssen wir zudem Folgendes berücksichtigen:

* Die Methode *checkIn()* gibt die Zimmernummern eines freien Zimmers zurück. Sind keine freien Zimmer verfügbar, liefert der Methodenaufruf 0 zurück.
* Mit der Methode *checkOut()* geben wir ein belegtes Zimmer wieder frei. Bitte darauf achten, dass nur belegte Zimmer freigegeben werden können.

[](file:///C:\schule\unterricht\web\runestone_php_oop\build\php_oop\_images\image18.png)