

Aufgaben OOP mit PHP

- Erstellen Sie ein neues Projekt in Ihrer Entwicklungsumgebung mit dem Bezeichner php_oop
- Jede Aufgabe ist in einem einzigen File zu schreiben. Beispiel: Sämtlicher Code zu Aufgabe 1 befindet sich in dem File aufgabe1.php, Aufgabe 2 in aufgabe2.php,
- Wenn Sie mit den Aufgaben fertig sind, veröffentlichen Sie diese in einem public Repository auf Github und senden Sie mir den Link per E-Mail zu.
- Wird von einem/einer Teilnehmer*in keine Lösungen eingereicht, entspricht dies einem Fehltag.

Aufgabe 1: Klassen und Objekte

Erstellen Sie eine Klasse namens "Bicycle" mit den folgenden Eigenschaften:

```
$manufacturer  
$model  
$year
```

Erstellen Sie ein Objekt der Klasse "Bicycle" und setzen Sie die Werte der Eigenschaften auf "Kalkhoff", "Image 3.B Move Wabe" und 2022. Verwenden Sie dann echo, um die Marke, das Modell und das Jahr des Fahrrads anzuzeigen.

Aufgabe 2: Kapselung

Erstellen Sie eine Klasse namens "Person" mit den folgenden privaten Eigenschaften:

Hinweis: <https://www.informatikzentrale.de/zugriffsmodifikatoren-public-private-protected.html>

```
$name  
$age
```

Erstellen Sie öffentliche Getter- und Setter-Methoden für die Eigenschaften.

Hinweis: <https://stackoverflow.com/questions/4478661/what-are-the-advantages-of-using-getters-and-setters-instead-of-functions-or-sim>

Erstellen Sie ein Objekt der Klasse Person und setzen Sie die Werte der Eigenschaften mit Hilfe der Setter-Methoden. Verwenden Sie dann echo, um den Namen und das Alter der Person mithilfe der Getter-Methoden anzuzeigen.

Alle folgenden Aufgaben ohne Getter- und Setter-Methoden bearbeiten.

Aufgabe 3: Vererbung

Erstellen Sie eine Klasse namens "Person" mit den folgenden Eigenschaften

\$name
\$age
\$gender

Erstellen Sie eine Unterklasse namens "Student", die die Klasse Person erweitert und die folgenden Eigenschaften hinzufügt:

\$studentID
\$course

Erstellen Sie eine weitere untergeordnete Klasse mit dem Namen "Teacher", die die Klasse Person erweitert und die folgenden Eigenschaften hinzufügt:

\$teacherID
\$subject

Erstellen Sie Konstruktoren für alle Klassen, die ihre jeweiligen Eigenschaften initialisieren.

Hinweis: <https://www.php.net/manual/de/language.oop5.decon.php>

Erstellen Sie in jeder Klasse eine Methode namens "displayInfo", die die Eigenschaften des Objekts in einer formatierten Zeichenkette anzeigt.

Erstellen Sie ein Objekt der Klasse Person, ein Objekt der Klasse Student und ein Objekt der Klasse Teacher. Werte für die Eigenschaften können Sie sich ausdenken. Rufen Sie die Methode displayInfo() für alle Objekte auf.

Aufgabe 4: Polymorphie

Erstellen Sie eine Klasse namens "Shape" mit der folgenden Methode:

Hinweis: Sie können, müssen jedoch nicht den Datentyp festlegen. Ich zeige Ihnen im Unterricht wie das genau funktioniert.

getArea() : float

Erstellen Sie Unterklassen mit den Namen "Rectangle" und "Circle", die die Klasse Shape erweitern und die Methode getArea() überschreiben.

Fügen Sie in der Klasse "Rectangle" die folgenden Eigenschaften hinzu:

\$width (Float)
\$height (Fließkomma)

Fügen Sie in der Klasse Circle die folgende Eigenschaft hinzu:

\$radius (Float)

Berechnen Sie in der Methode getArea() der Klasse Rectangle den Flächeninhalt des Rechtecks und geben Sie ihn mit der folgenden Formel zurück: $\$width * \$height$.

Berechnen Sie in der Methode getArea() der Klasse Circle den Flächeninhalt des Kreises mit der Formel: $\pi * (\$radius ^ 2)$ und geben Sie ihn zurück.

Erstellen Sie ein Objekt der Klasse Rectangle mit einer Breite von 10 und einer Höhe von 5. Erstellen Sie ein Objekt der Klasse Circle mit einem Radius von 4. Verwenden Sie echo, um den Flächeninhalt des Rechtecks und des Kreises anzuzeigen.

Aufgabe 5: Abstraktion

Erstellen Sie eine abstrakte Klasse mit dem Namen "Animal" mit den folgenden abstrakten Methoden.

Hinweis: <https://www.php.net/manual/de/language.oop5.abstract.php>

makeSound() : string
eat(string \$food) : void

Erstellen Sie Unterklassen namens "Hund" und "Katze", die die Klasse "Animal" erweitern und die Methoden makeSound() und eat() implementieren.

Geben Sie in der Methode makeSound() der Klasse Dog die Zeichenkette "Woof!" zurück. Die makeSound()-Methode der Klasse Cat gibt die Zeichenkette "Meow!" zurück.

Geben Sie in der Methode eat() der Klasse Dog die Zeichenkette "The dog is eating " gefolgt vom Parameter food zurück. Geben Sie in der Methode eat() der Klasse Cat die Zeichenkette "The cat is eating" gefolgt vom Parameter food zurück.

Erstellen Sie ein Objekt der Klasse Hund und ein Objekt der Klasse Katze. Rufen Sie die Methoden makeSound() und eat() für beide Objekte auf.