

Aufgabe 2 - OOP

Wichtig:

Da wir einen Online-Compiler verwenden, werden wir alle unsere Klassen in einer einzigen Datei definieren. Damit dies funktioniert, müssen Sie die Klasse Main am Anfang der Datei einfügen, die anderen Klassen sollten danach folgen. Beispiel:

```
class HelloWorld {  
    public static void main(String[] args) {  
        Cat cat = new Cat();  
        cat.speak();  
    }  
}
```

```
class Cat {  
    public void speak() {  
        System.out.println("Meow!");  
    }  
}
```

Aufgabe 1 - Klasse Rectangle

- a. Erstellen Sie eine Klasse *rectangle* mit zwei Attributen
 - double width
 - double height
- a. Fügen Sie eine Methode namens `calculateArea()` hinzu, die die Fläche des Rechtecks zurückgibt.
- b. Erstellen Sie in der Methode `main()` eine Instanz der Klasse *rectangle*, weisen Sie ihren Feldern Werte zu und geben Sie die Fläche mit der Methode `calculateArea()` aus.

Aufgabe 2 - Klasse Circle

- c. Erstellen Sie eine Klasse *Circle* mit einem Attribut
 - double radius
- d. Fügen Sie eine Methode namens `calculateArea()` hinzu, die die Fläche des Kreises zurückgibt.
- e. Erstellen Sie in der `main()`-Methode eine Instanz der Klasse *Circle*, weisen Sie ihren Feldern Werte zu und geben Sie die Fläche mit der Methode `calculateArea()` aus.

Bonus Aufgabe - Interface gestalten

Ist es möglich, *Rectangle* und *Circle* zusammenzufassen?

- f. Erstellen Sie eine Shape-Schnittstelle mit einer Methode namens `calculateArea()`.
- g. Implementieren Sie die Schnittstelle mit *Rectangle* und *Circle*.

- h. Erstellen Sie in der `main()`-Methode ein Array von `Shape`, das Rechtecke und Kreise enthält, und addieren Sie deren Flächen mit einer `for`-Schleife.