

# ÜBUNGSSERIE 3

## Algorithmen & Datenstrukturen AD1 / FS 2019

AD1 Team

### Aufgabe 1 (O-Notation)

Problem: PAAR-Summen:

Eingabe: Eine Folge von  $n$  verschiedenen Zahlen  $a_1, \dots, a_n$ .

Ausgabe: Alle paarweisen Summen  $a_i + a_j$ , für  $1 \leq i < j \leq n$ , d.h. für alle Paare von Zahlen aus der Eingabe deren Summe

Beispiel:

```

a1..an:  47, 73, 21, 17, ...
i=1 ->  47+73, 47+21, 47+17, ...
i=2 ->           73+21, 73+17, ...
i=3 ->                21+17, ...
...
i=n-1                an-1+an

```

- Entwerfen Sie einen Algorithmus, der dieses Problem löst.
- Bestimmen Sie die Laufzeit des Algorithmus (in expliziter Form (als Polynom) und in der O-Notation).
- Implementieren Sie den Algorithmus (siehe *PairSums.java*)
- Optional: Führen Sie eine Laufzeitmessung durch (siehe Methode `runtimeMeasurement()`)

### Aufgabe 2 (Vollständige Induktion)

*Hinweis: Beachten Sie die Unterlagen auf dem Skripte-Server in 7\_Zusatzmaterial\Vollstaendige Induktion*

- Beweisen Sie mit Hilfe der vollständigen Induktion die explizite Summenformel aus Übung 1 / Aufgabe 4 (a).
- Welche Analogie besteht zwischen dem Beweisverfahren „Vollständige Induktion“ und der Rekursion?

### Aufgabe 3 (Vollständige Induktion)

Beweisen Sie mit Hilfe der vollständigen Induktion die explizite Summenformel aus Übung 1 / Aufgabe 4 (b).

## Aufgabe 4 (Adapter Pattern)

Sie haben die Aufgabe erhalten eine einfache Datenhaltung für einen Bike-Shop zu schreiben und gesehen, dass die Java-Klasse `ArrayList` alle Methoden implementiert, die Sie benötigen.

- a) Schreiben Sie einen Objekt Adapter *ObjAdapter*, der die *ArrayList* Klasse verwendet und die folgende Klasse *AbstractBikeStore* realisiert:

```
public abstract class AbstractBikeStore
{
    public abstract void addBike(Bike b);
    public abstract Bike removeBike(int i);
    public abstract void clear();
    public abstract int size();
}
```

- b) Erstellen Sie ein Interface *BikeStoreInterface*, welches die Methoden der abstrakten Klasse beinhaltet.
- c) Schreiben Sie nun auch einen Klassen Adapter *ClassAdapter*, der das Interface verwendet.

