

# Opdracht 2 motivering

## Opdracht

**\*Lever als concept enkel een schriftelijke beschrijving in van de datastructuren die je wilt gebruiken, en motivatie waarom je die hebt gekozen. Met andere woorden: welke interfaces en implementatie klassen ga je gebruiken en met welke actuele type parameters? Stuur dus als conceptversie geen code in.\***

### Welke implementatie klassen?

Voor opdracht 2 wil ik drie klassen maken Woord, WoordenLijst en SynoniemenLijst. Woord is heel simpel en heeft alleen een id en een string. Voor de Woorden in de WoordenLijst is ieder id unique. Voor de Woorden in de synoniemenlijst is dit geen gegeven, bij het toevoegen van synoniemen krijgen alle Woorden namelijk de id van het Woord uit de Woordenlijst waar ze bij horen.

### Welke interfaces, met welke actuele type parameters?

Dit zijn de interfaces in pseudo-code met de actuele type parameters in cursief:

---

#### Klasse Woord

- *int* id
- *String* woord

#### Klasse WoordenLijst

- *ArrayList* WoordenLijst<*Woord*>
  - *SynoniemenLijst* synoniemen
  - *int* size
- getWoord(int id) → \*Woord\**  
*checkDuplicates(String) → \*Boolean\**  
*addWoord(String)*  
*sort()*

#### Klasse SynoniemenLijst

- *ArrayList* SynoniemenLijst<*Woord*>
- getSynoniemen(int id) → [\*Woord\*]*  
*addSynoniemen( [String] )*  
*sort()*

---

## Motivatie:

Ik heb voor deze datastructuur gekozen om het zo simpel mogelijk te houden. Woorden zijn key value pairs, die keys (het woord) maak ik tot een array voor die JList om te tonen en bij het aanklikken vraag ik in de event de value oftewel de id op. In principe zou dit ook gewoon de string zelf kunnen zijn maar een getal voelt wel netter. Met die id loop ik door de Synoniemenlijst en toon ik alle synoniemen met de corresponderende id.

Bij het toevoegen van een woord check ik eerst of deze voorkomt en geef ik hem dan de size + 1 als id mee. Die size hou ik bij omdat dat goedkoper is bij hele grote Woordenlijsten.

Die Woordenlijst is de Kapstokklasse omdat deze ook een compositie heeft met de Synoniemenlijst. Dit had ook in een overkoepelende klasse gekund, maar aangezien synoniemen bij een woord horen leek het mij logischer om de Woordenlijst de kapstok te maken.

Ik heb het gevoel dat het ingewikkelder moet en dat ik wellicht een nieuw type lijst moet maken die niet alle functionaliteiten van ArrayList biedt (zoals remove). Maar gezien de klassen toch geëncapsuleerd worden is het volgens mij ook wel overbodig om dit te doen. 🤔