**Bij welke methodes in Kassa en Kantine komt dit voor?**

resetKassa

hoeveelheidGeldInKassa

aantalArtikelen

**Leg uit waarom het goed is om de methodes ArrayList<Artikel> getArrayList(String productnaam) en Artikel getArtikel(ArrayList<Artikel>) private te maken.**

Deze methodes worden gebruikt in de klasse en niet buiten de klasse, het is dus ook niet nodig om dit zichtbaar te laten zijn. Vandaar de private.

**In welke situatie gebruik je een HashMap en wanneer een HashSet?**

HashMap en HashSet lijken veel op elkaar als je naar het woordt kijkt. Ze verschillen echter wel veel van elkaar als je kijkt hoe ze beiden werken.

Een HashMap gebruik je om met een sleutel de verbonden waarde op te halen.

Een HashSet gebruikt geen sleutels maar een index, net als een ArrayList. Verder werkt een HashSet ook in andere opzichten als een ArrayList- behalve dat alle waarden maar een keer mogen voorkomen, Elke waarde is dus uniek

**Leg de werking van de constructor uit.**

Zonder kantine kun je niets simuleren, dus eerst maken we deze aan: *new Kantine.*

Daarna maken we random aan omdat we niet elke dag dezelfde simulatie willen zien.

*random = new Random;* is nodig om daarna de random methodes aan te roepen.

*Int[] hoeveelheden* wordt aangemaakt omdat er een array nodig is voor de voorraad.

Elk artikel heeft een voorraad.

De parameter aantalArtikelen bepaalt hoe lang de *int[] hoeveelheden* array wordt.

Dit is voor nu 4 🡺 *private static final int AANTAL\_ARTIKELEN=4;*

*kantine.setKantineAanbod(artikelnamen, artikelprijzen, hoeveelheden);*

Hier wordt een methode aangeroepen van de Klasse Kantine die op zijn beurt de constructor van de klasse KantineAanbod aanroept.

Na het uitvoeren van de net genoemde constructor zijn er 4 artikelen met bijbehorende prijs en voorraad aangemaakt.

**Leg de implementatie van int getRandomValue(int min, int max) uit en met name waarom er +1 in voorkomt. Gebruik de Java API. Hint: denk aan de betekenis van inclusief en exclusief.**

getRandomValue() geeft een willekeurig getal terug.

De minimumwaarde wordt van het maximumwaarde afgetrokken en hier wordt 1 bij opgeteld.

Er 1 bij optellen moet omdat nextInt() een getal kiest tussen 0 (nul) en het maximum, waarbij 0 inclusief is en het maximum exclusief.

Omdat we in de klasse KantineSimulatie ervan uit gaan dat het maximum inclusief is, zal er dus 1 bij het maximum opgeteld moeten worden. Anders zal het maximum nooit voorkomen in de output van de methode.