

## Oefening 1: Breuken

Ontwerp een klasse Breuk met twee private attributen voor de teller en de noemer.

Voorzie twee constructoren, een eerste zonder argumenten en een tweede met twee argumenten, namelijk de waarden voor teller en noemer.

Voorzie in de klasse drie getters respectievelijk om de teller, de noemer en de rationale waarde van de breuk op te vragen.

Voorzie twee setters respectievelijk voor de teller en de noemer.

Ontwerp een toString methode zodat een Breuk-object mooi en gemakkelijk geprint kan worden ( vb. in de vorm teller/noemer)

Voorzie een *plus* methode met één argument nl. een Breuk-object: een nieuw Breuk-object wordt gemaakt op basis van de som van de breuk waarop de methode toegepast wordt en de breuk die als argument meegegeven is.

Voorzie op analoge manier een *maal*-methode om 2 breuken met elkaar te vermenigvuldigen.

Voorzie een tweede *maal*-methode om een breuk te vermenigvuldigen met een geheel getal; het getal wordt als argument meegegeven.

Test deze klasse uit in een main-methode (deze mag in een nieuwe klasse geschreven worden zodat de gebruikte toegang duidelijk wordt):

- maak twee breuken: vb.  $1/2$  en  $-3/4$
- druk de breuken af op het scherm
- bereken de rationale waarde van de breuken en druk ze af
- tel de breuken op en druk de resulterende breuk af in breuk-notatie en als rationeel getal
- vermenigvuldig de breuken en druk de resulterende breuk af
- trek de breuken van elkaar af (door eerst te vermenigvuldigen met  $-1$ ) en druk de resulterende breuk af.
- Maak een nieuw Breuk-object dat verwijst naar een reeds bestaande breuk (dit doe je met de `=` operator) en wijzig de teller en/of noemer van de originele breuk. Controleer wat er gebeurt met de nieuwe breuk.

### Uitbreiding:

Schrijf 2 private methodes *void vereenvoudig()* en *int ggd(int, int)* die ervoor zullen zorgen dat een breuk altijd in zijn vereenvoudigde vorm wordt opgeslagen. Test deze methodes uit.

Maak in je main-methode een array van 5 *Breuk*-objecten waarin je de aangemaakte breuken bewaart. Test met deze array-elementen opnieuw de optelling, de vermenigvuldiging en de aftrekking en bewaar de resultaten ook in de array. Druk op het einde de volledige array af. (zie paragraaf 2.4 en 3.2.3 in de cursus)