Lessen Java: Reeks 2

David Blinder
Jan G. Cornelis

Wat is een klasse (class) ?

Alles wat geen **primitief type** (int, double, boolean, enz.) is, is een klasse.

Klassen bevatten **attributen** en **methodes**:

- **Attributen** zijn de onderdelen van de klasse. Attributen kunnen zelf klassen zijn of primitieve types.
- **Methodes** zijn functies die gerelateerd zijn aan de Klasse.

Primitieve Types

type	grootte	waardes
byte	8 bits	$[-128, 127]$
short	16 bits	$[-32768, 32767]$
int	32 bits	$[-2^{31}, 2^{31} - 1]$
long	64 bits	$[-2^{63}, 2^{63} - 1]$

type	grootte	waardes
float	32 bits	$\approx$ 7 bed. cijfers
double	64 bits	$\approx$ 16 bed. cijfers
boolean	1 bit *	$\{false, true\}$
char	16 bit	‘a’, ‘R’, ‘7’, ‘?’, …

- `Klassen zijn opgebouwd uit andere klassen en/of primitieve types`
- **`String`** is geen primitief type: Bestaat intern uit een array van **`char`**
- Primitieve types zijn gekleurd + boldface in Eclipse, en bevatten geen methodes.

Wat is een klasse ?
(bv. bevolkingregister)

Class Persoon

```
String voornaam
String achternaam
Geslacht geslacht
Datum geboortedatum
Adres woonplaats
Adres werkplaats
…
```

ATTRIBUTEN

```
void print_gegevens(int paginas)
int bereken_leeftijd()
```

METHODES

Class Adres

```
String Land
String Plaats
int huisnummer
String Straat
…
```

ATTRIBUTEN

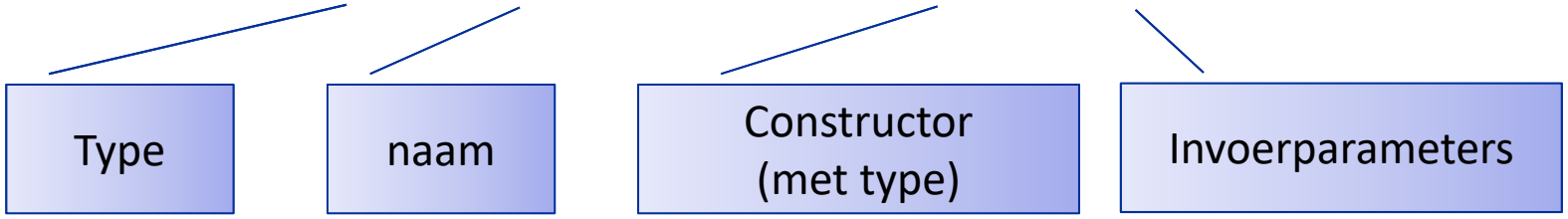
```
void print_adres()
…
```

METHODES

Aanmaken van een object

Het aanmaken van een nieuw object doe je doormiddel van het keyword **new** en de constructor van die klasse:

```
Klasse obj = new Klasse(a,b,...);
```



Klassen kunnen meerdere constructors hebben

Constructor mag van een subtype zijn. Voorbeeld:

```
Breuk x = new Breuk(1,4);
Persoon jef = new Student("Jef", 1995, "Sociologie");
```

Random (vraag 1)

De klasse Random laat je willekeurige getallen genereren.

```
import java.util.Random;
```

```
Random rand = new Random();
```

Aantal nuttige methodes:

- **nextBoolean**() — Geeft een **true** of **false** terug met 50/50 kansen
- **nextInt**(int n) — Geeft een int terug tussen 0 en $n-1$ (uniform verdeeld)
- **nextDouble**() — Geeft een double terug uit $[0,1[$ (uniform verdeeld)
- **nextGaussian**() — Geeft een double terug uit een Gaussiaanse verdeling met $\mu = 0$ en $\sigma = 1$

Arraylist (vraag 1)

```
import java.util.List;
import java.util.ArrayList;

ArrayList<int> list = new ArrayList<int>();
ArrayList<Integer> list = new ArrayList<Integer>();
```

Kies het type voor **T** (jokerteken)

```
ArrayList<T>
        void add(T)          int size()
        T get(int)           T remove(int)
```

Nuttige methodes: (zie ook http://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html en http://parallel.vub.ac.be/education/java/miniJava/classes/Arraylist.html)

Arrays (vraag 2)

Arrays (i.t.t. ArrayLists) hebben een fixe lengte bij creatie. Je kan ook primitieve types gebruiken. Geschikt voor structuren zoals afbeeldingen.

Nuttig voor meerdimensionale constructies

```
double[] arr1 = new double[10];
        // array van doubles (10 elementen)
arr1[2] = 17.5;
        // derde element toekennen
boolean[][] arr2 = new boolean[3][5];
        // 2D array van booleans (3x5 elementen)
int[] arr3 = {1,2,3,4,5};
        // array van ints + initialisatie van waardes
arr3.length → 5
        // geeft lengte terug
```

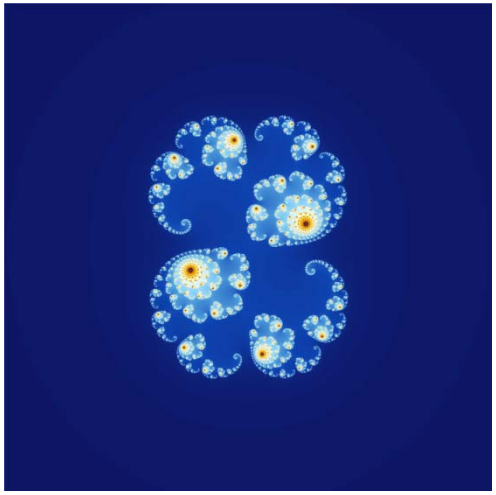
Juliafractaal (vraag 2)

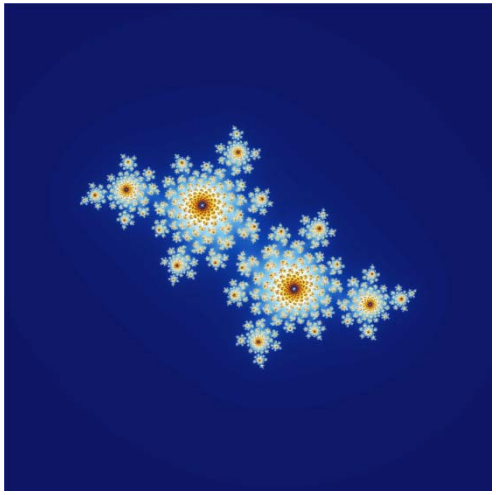
Een fractaal is een meetkundige figuur die zelfgelijkend is*.

(een deel van) de Julia sets worden beschreven door:

Convergeert de reeks $\lim_{i\to\infty} z_i(x)$? $z_{n+1}(x) = z_n^2(x) + c$ & $z_0(x) = x$

Constante c, bepaal convergentie $\forall x \in \mathbb{C}$





Moeilijk te bepalen. Praktische benadering: kies een $M > 0$, reken uit na hoeveel (k) stappen de reeks lijkt te divergeren $|z_k(x)| > M$

Juliafractaal (vraag 2)

Twee meegegeven klassen (zie parallel-website):

1. **ComplexNumber:** stelt een complex getal voor
2. **SimpleImage:** laat toe eenvoudige afbeeldingen weer te geven.

Bestudeer de meegegeven klassen, gebruik deze om in een nieuwe klasse **JuliaMain** waar de main-methode staat.

Kies als constanten:

- $c = -0.70176 - 0.3842\text{i}$
- $M = 5$
- Test iteraties uit voor k tot maximaal 255.

Klasse Breuk (vraag 3)

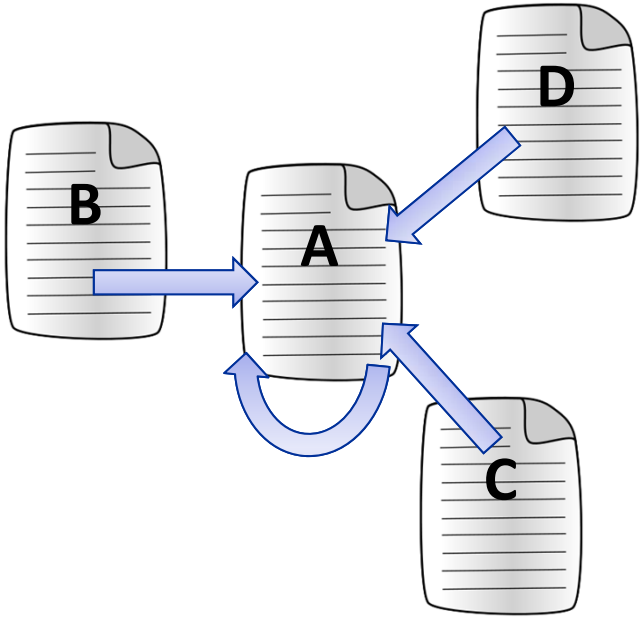
```
public class Breuk {
    private int teller, noemer;

    public Breuk(int t, int n) {
                teller = t;
                noemer = n;
    }

    public void optellen(Breuk other) {
        teller = this.teller*other.noemer + other.teller*this.noemer;
        noemer = this.noemer*other.noemer;
    }

    public String toString() {
        return teller+"/"+noemer;
    }
}
```

VRIJE UNIVERSITEIT BRUSSEL

Encapsulatie: public en private

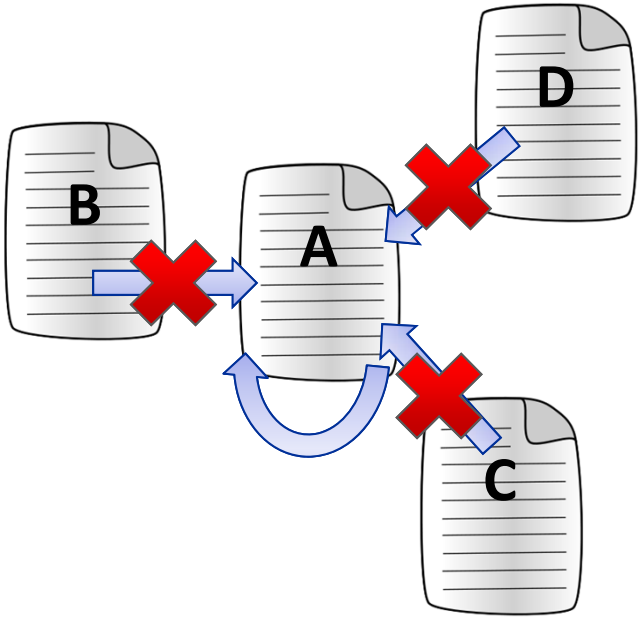
```
public class A
{
    public int x;
…
```

Public




```
public class A
{
    private int x;
…
```

Private



Klasse Test (bevat de main-methode)

```
public class Test {

        public static void main(String[] args) {
                Breuk a = new Breuk(1,2);
                Breuk b = new Breuk(1,3);

                System.out.print(a + " + " + b + " = ");
                a.optellen(b);
                System.out.println(a);
}
```

Output: 1/2 + 1/3 = 5/6

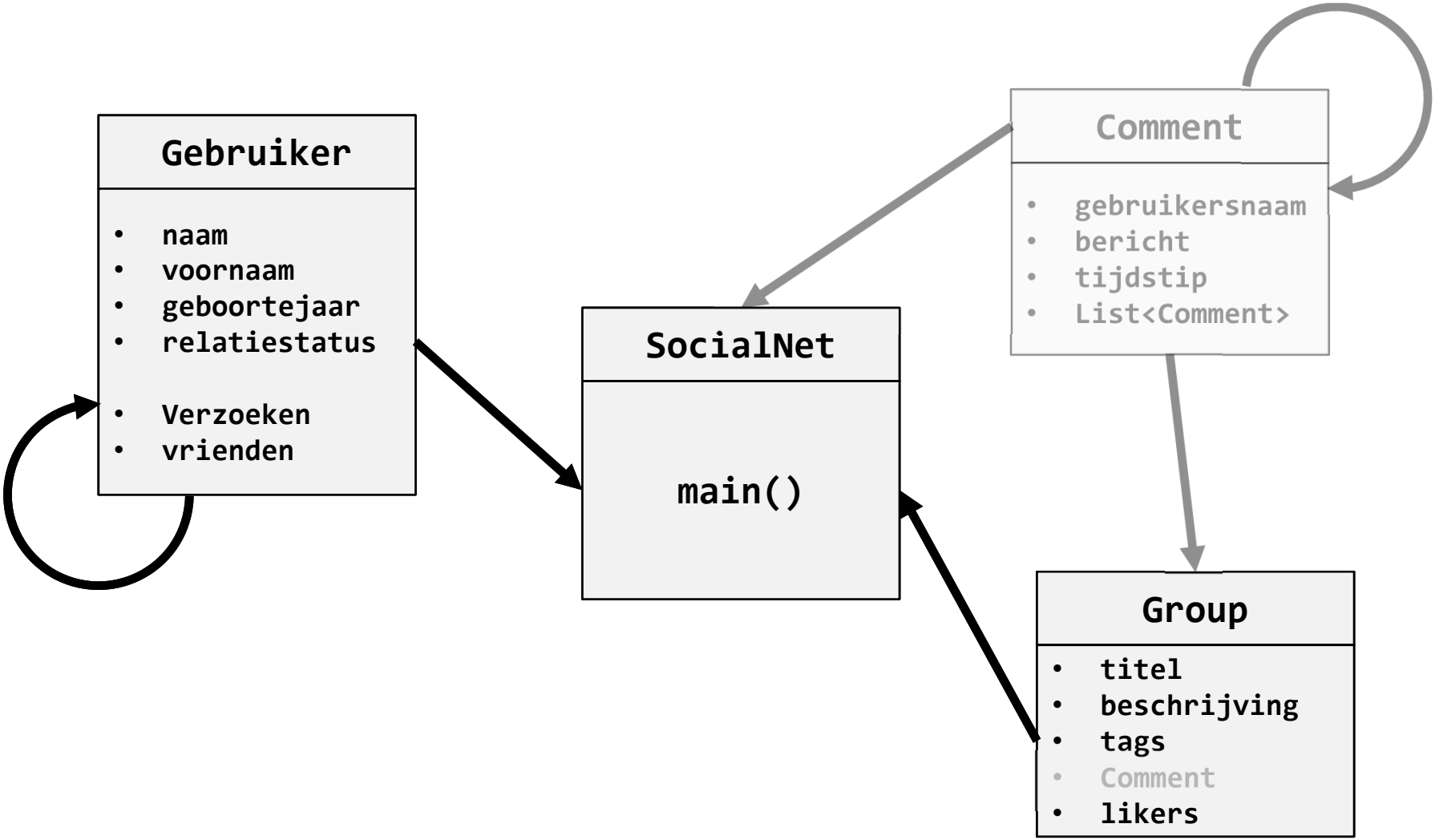
Greatest Common Denominator
(Grootste gemeenschappelijke deler)

- Algoritme van Euclides:

$$\gcd(a,b) = \begin{cases} a > b & \gcd(a-b,b) \\ a = b & a \\ a < b & \gcd(a,b-a) \end{cases}$$

- Zorg dat de **“normalize()”** methode telkens wordt opgeroepen wanneer de **Breuk** wordt aangepast.

SocialNet


Gebruiker
naam
voornaam
geboortejaar
relatiestatus
Verzoeken
vrienden
SocialNet
main()
Comment
gebruikersnaam
bericht
tijdstip
List<Comment>
Group
titel
beschrijving
tags
Comment
likers

Klasse - speciaal geval: ENUM

```
public enum Weekdag {
        Maandag, Dinsdag, Woensdag, Donderdag, Vrijdag,
        Zaterdag, Zondag;
}

public enum Kleur {
        Rood, Geel, Groen, Blauw;
}

Kleur k = Kleur.Rood;
```

Aanmaken: **New → Enum**

Set

Set<E>: een collectie van elementen die geen duplicaten bevat (gelijkaardig aan de mathematische definitie van een verzameling).

Voorbeelden:

- **Vrienden:** Set<Persoon>
- **Media:** Set<Film>

Nuttige methodes:

- add(T) $A \coloneqq A \cup \{a\}$
- remove(T) $A \coloneqq A \setminus \{a\}$
- int size()
- boolean contains(T)

Map

Map<K, V>: een structuur die keys (van klasse **K**) “mapt” op values (van klasse **V**).

Voorbeelden:

- **Inventaris:** Map<Item, Integer>
  (Object → Aantal)
- **Telefoonboek:** Map<String, String>
  (Persoon → Telefoonnummer)

Nuttige methodes:

- V get(K)
- void put(K, V)
- int size()
- Set<K> keySet()

List<A> = new ArrayList<A>

Map<A,B> = new HashMap<A,B>

Strings vergelijken

Strings vergelijken met **equals()**, niet met == !

```
String x = "abcdef";
String y = "abcdef";
String z = "abc";

……

if (voorwaarde) {
        z += "def"
}

x == y; // true
x == z; // false !
```