examen java oo

# HFDST 1: Introductie

Console toepassingen in Java

****

Javadoc verwijzing naar andere methoden via @see

Statische constante

= een veld dat door alle objecten van de klasse gedeeldwordt én slechts 1 keer een waarde kan krijgen, ofwel bij declaratie ofwel in de constructor.

Standaard en niet standaard constructor

Bij aanwezigheid van een niet standaard constructor wordt GEEN standaard constructor meer gegenereerd. Wil je die ook, dan moet je die expliciet schrijven.

# HFDST 2: Packages - Exception Handling – Enumeraties

Copy-constructor

MERK OP: Initialisatie van de velden via <veldnaam>of this.<veldnaam>

Exception Handling

•A. In methode of Constructor

Via **if/else** vb:

public void setLeeftijd(int leeftijd) {

if(leeftijd < 0) {

throw new IllegalArgumentException("leeftijd mag niet negatief zijn");

}

this.leeftijd= leeftijd;

}

•B. In presentatie

Via **try/catch** vb:

Try{

Persoon p = new Persoon();

p.setNaam("Van Assche”);

p.setVoornaam(“Kristien");

p.setLeeftijd(-11);

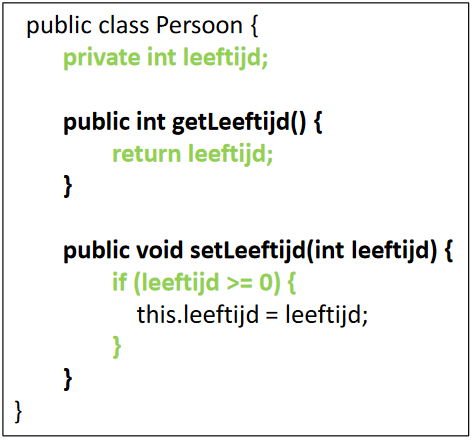
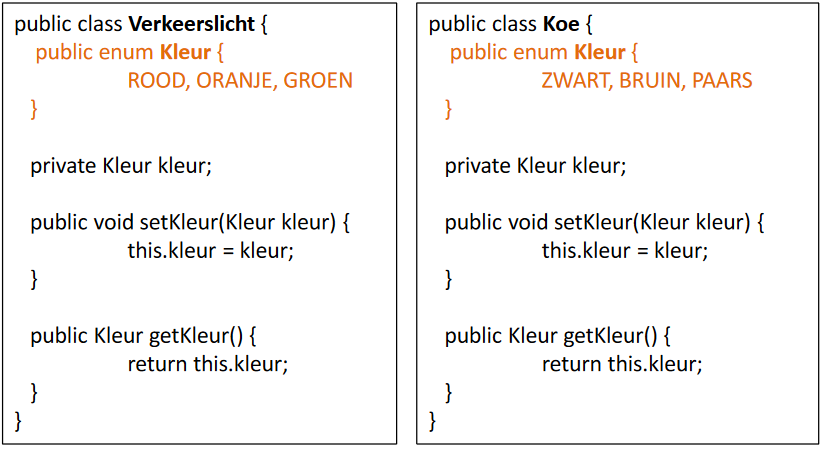
p.setGeslacht(true);

} catch (IllegalArgumentException e) {

System.out.println("FOUT:" + e.getMessage());

}

Enumeraties

****

Privaat veld met ‘gecontroleerde’ toegang via getter-en setter methode

# HFDST 3: Excepties - Enumeraties – GUI

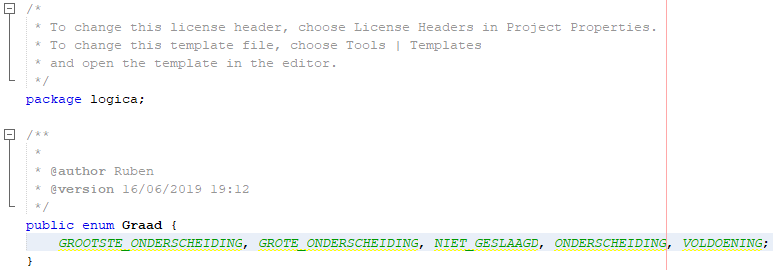
Algemene exception afhandelen

****Enumeraties in aparte package onderbrengen

# HFDST 4: GUI – Netbeans

# LABO 5:

Enumeratie als klasse aanmaken

Bep****aalt aantal getallen na de komma

DecimalFormat df= new DecimalFormat("0.00");

this.jLabelOutput.setText("Afstand: " + df.format(afstand));

GUI achtergrondkleur/visibility instellen

this.jTextFieldXXX.setBackground(Color.red);

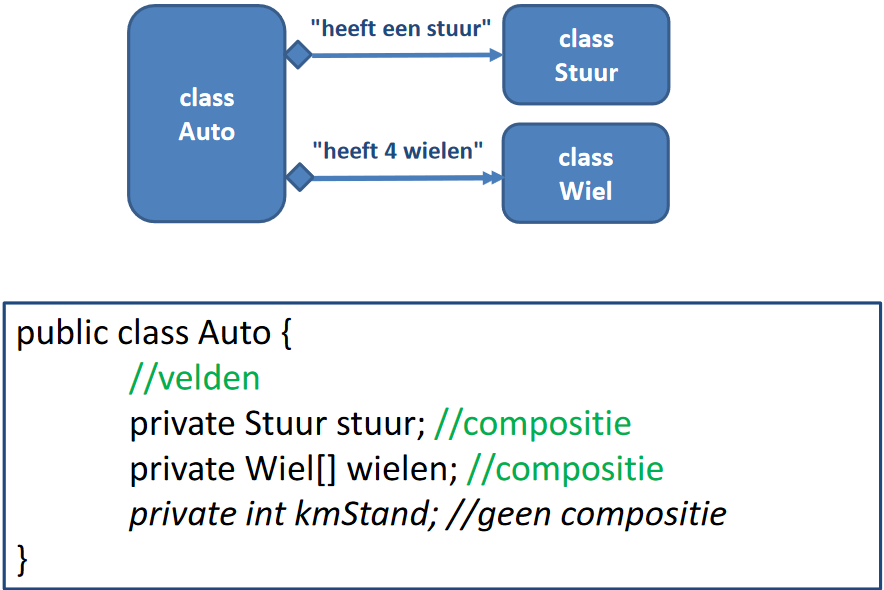
this.jTextFieldXXX.setBackground(Color.green);

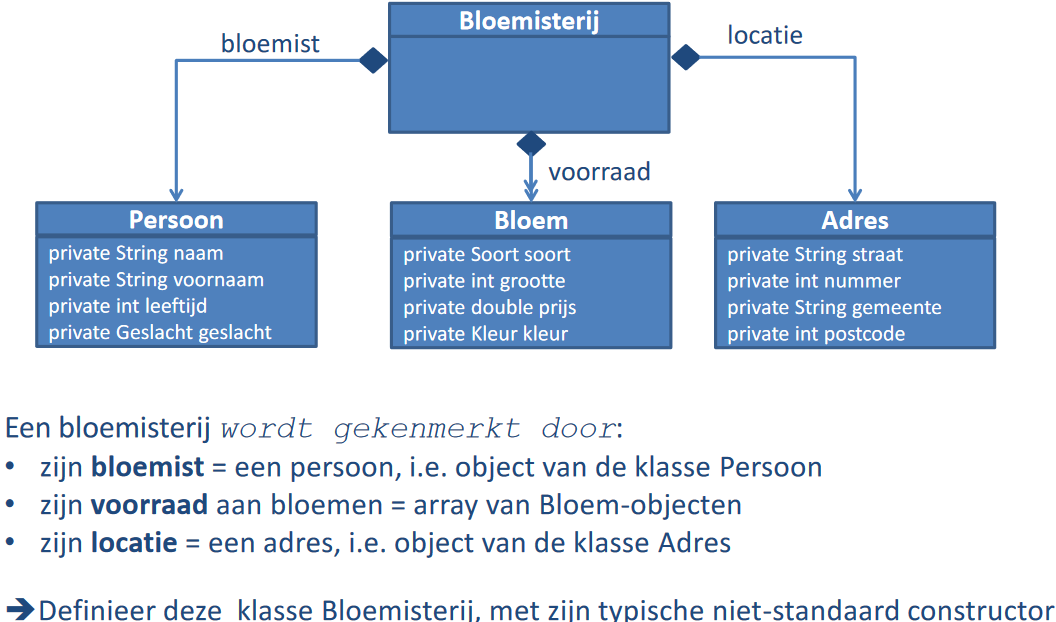
this.jTextFieldXXX.setBackground(Color.orange);

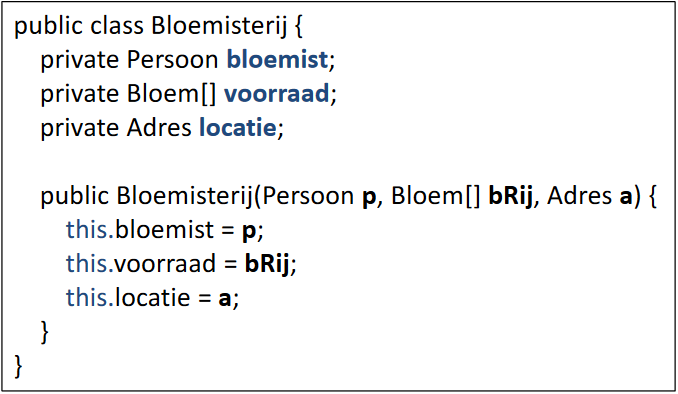
this.jLabelXXX.setVisible(false);

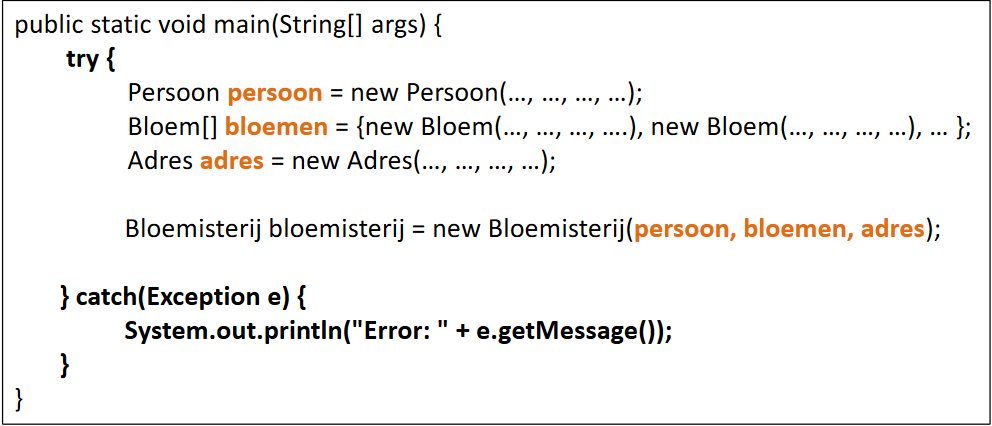
this.jTextFieldXXX.setVisible(false);

# HFDST 5: compositie & overerving

UML – Compositie relatie = "heeft een" relatie

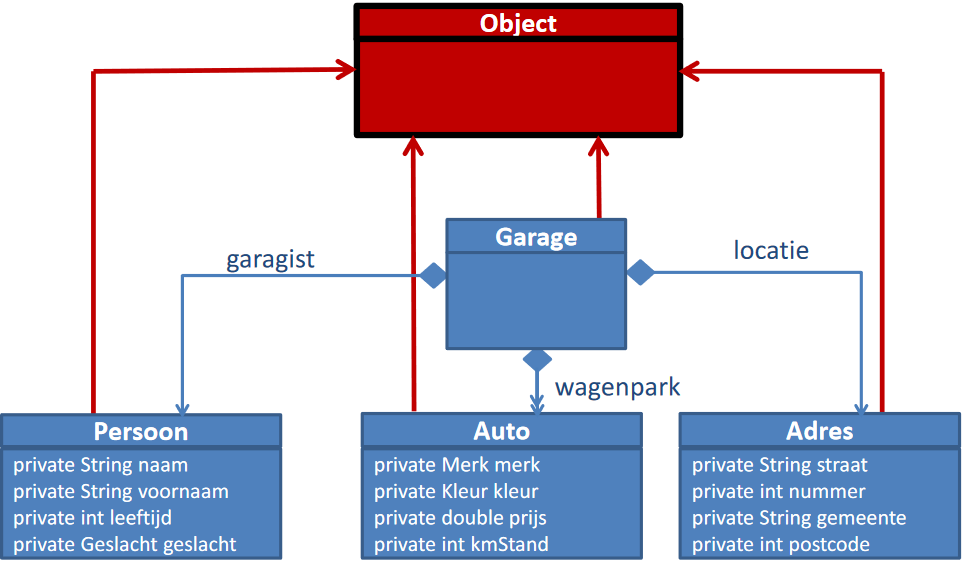
Uitgebreider voorbeeld:



Hoe maak je dan een concreet object van deze klasse Bloemisterij?

OVERERVING = "erft van" relatie

ALLE klassen erven over van dé root klasse Object



Wát erf je dan?

Kenmerken en functionaliteit v/d basisklasse

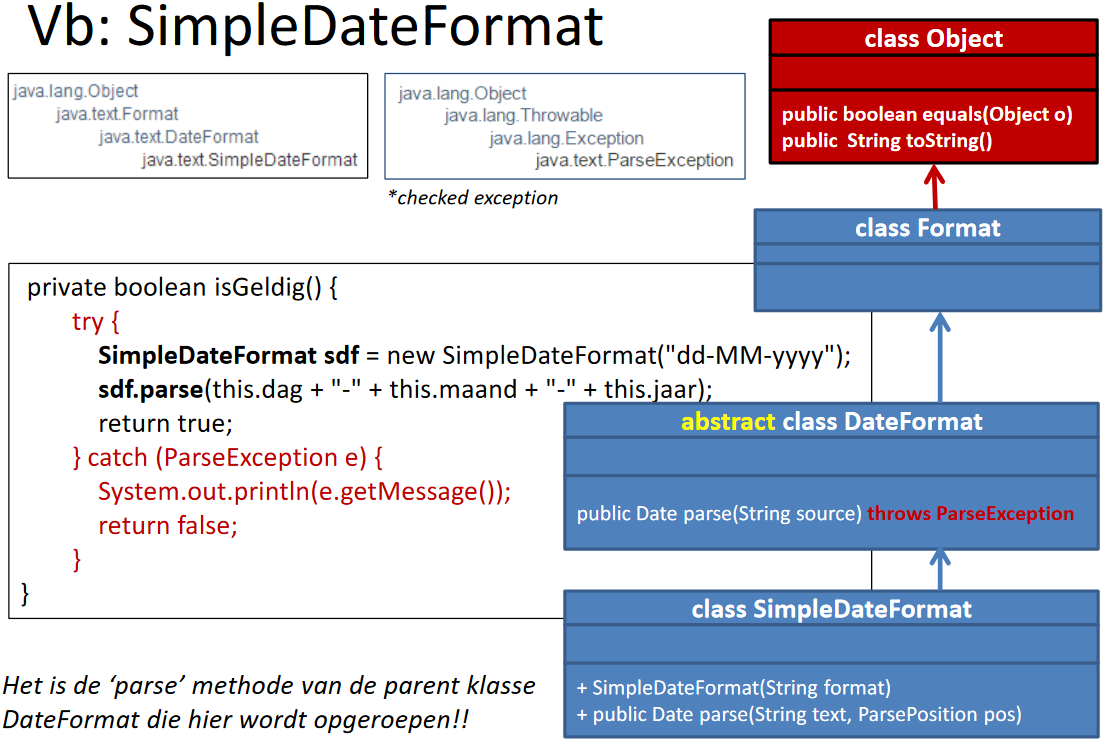
• Kenmerken 🡪 velden

• Functionaliteit 🡪 methoden

De ‘root’ methoden (en hun vaste signatuur!)

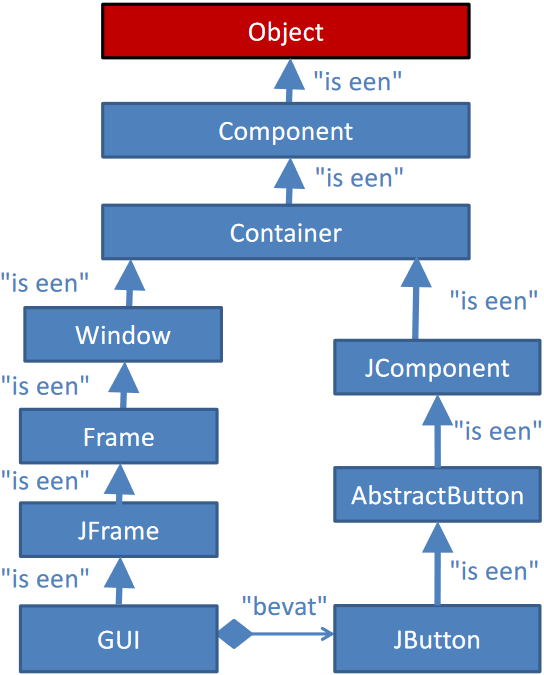
•toString public String toString()

•equals public boolean equals(Object o)

abstracte klasse

Geen objecten/instanties mogelijk van deze klasse!

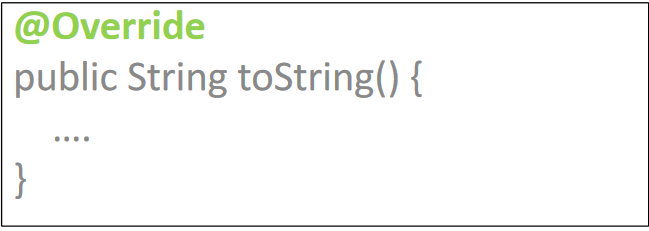
GUI-venster & knop (UML)



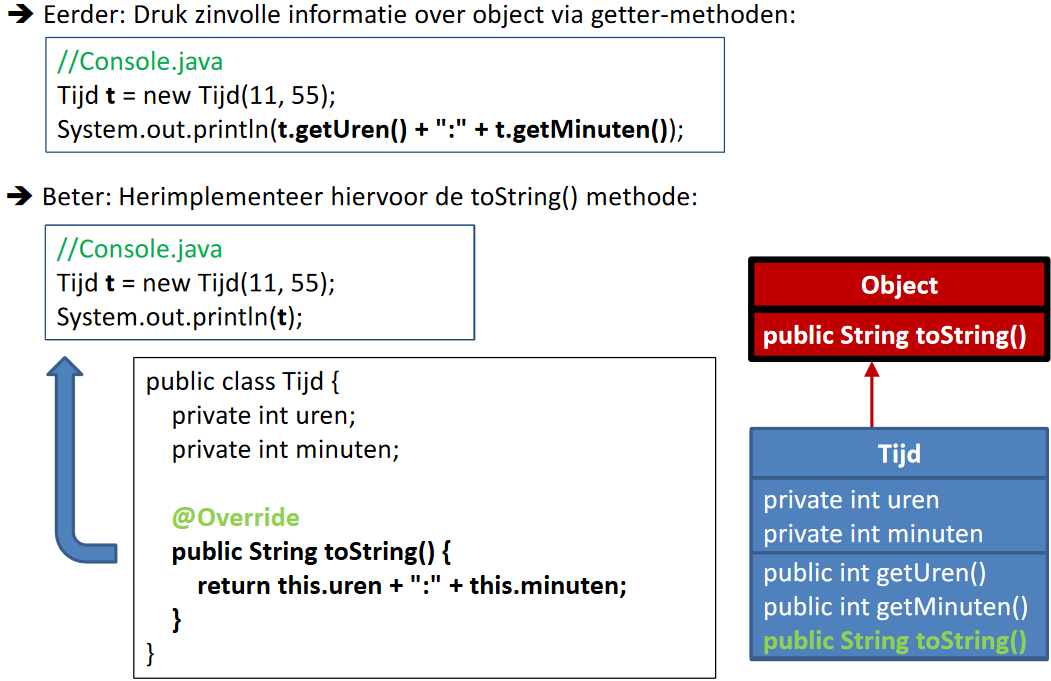
class X extends Y

"To extend" betekent letterlijk "uitbreiden":

In de subklasse kan je de set van velden en methoden uit de basisklasse "uitbreiden" met extra velden en methoden.

****Je kan ook bestaande methoden een andere invulling (i.e. andere implementatie) geven.

Herimplementatievan de toString() methode



Voorbeeld: Tijd