

Inhoudsopgave: Datum:

# Overerving Deel 2

## Herhaling + Verduidelijking

Schema van alle mogelijk datatypes:

Afbeelding met tekst, schermopname, lijn, Lettertype

Automatisch gegenereerde beschrijving

Bij het maken van objecten/referenties van variabelen komen, zoeken deze waarden plaats op in het **RAM** en wordt het daar opgeslagen.

Communiceren met objecten via de *de-reference* operator:

Om te spreken met objecten gebruiken we altijd een **.** om dit te doen.

Afbeelding met tekst, schermopname, diagram, lijn

Automatisch gegenereerde beschrijvingIS-EEN Relatie (BOTTOM TO TOP):

Een overerving van een superklasse is altijd een **IS-EEN relatie**.

* Rechtoek **IS EEN** figuur.
* Driehoek **IS EEN** figuur.

KAN EEN Relatie (TOP TO BOTTOM):

* Object **KAN EEN** Figuur zijn.
* Rechthoek **KAN EEN** Vierhoek zijn.

Alle soorten verbindingslijnen in Visual Paradigm:

Afbeelding met tekst, schermopname, lijn, diagram

Automatisch gegenereerde beschrijving

Afbeelding met schermopname, tekst, Lettertype

Automatisch gegenereerde beschrijving

SUPER-klasse:

* De superklasse wordt **altijd** aangeroepen, of je het er nu bijschrijft of niet, hij staat er altijd.
* Is het geen superklasse van overerving, wordt het wel altijd aangeroepen vanuit de klasse **Object**.

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

Visibility protected:

Dit attribuut kan enkel gebruikt worden in **dezelfde** **package** of Een Superklasse ervan.

Default visibilty Java:

Packagewidevisibility

Instanceof:

Afbeelding met tekst, Lettertype, schermopname

Automatisch gegenereerde beschrijving

## Klasse Object

Object methoden:

* Equals: definieert de notie van **gelijkheid** gebaseerd op de toestand van een object, los van de **referentie**.
* Tostring: geeft een **string representatie** terug van het object.
* Getclass: geeft een **referentie** terug naar het **class object**, een representatie van de klasse van dat object.

Abstracte klassen:

* Je geeft aan dat het niet de bedoeling is om **instanties** te maken van de **superklasse**, enkel van de **subklassen**, maar je kan wel zeggen **Rekening rekening = new Zichtrekening** (geen rekeningen, maar spaar- of zichtrekening).
* Abstracte klassen hebben ook **geen** **body** in de superklasse.
* 1 abstracte methode = volledige klasse abstract
* Abstracte klassen/methoden in **UML** = **schuingedrukt**.
* **Dit is de reden waarom we de abstracte methode in superklasse Rekening hebt gezet.**

Afbeelding met tekst, Lettertype, Multimediasoftware, schermopname

Automatisch gegenereerde beschrijving