Zandpoort

6A&D
Software Development

UML-notatie

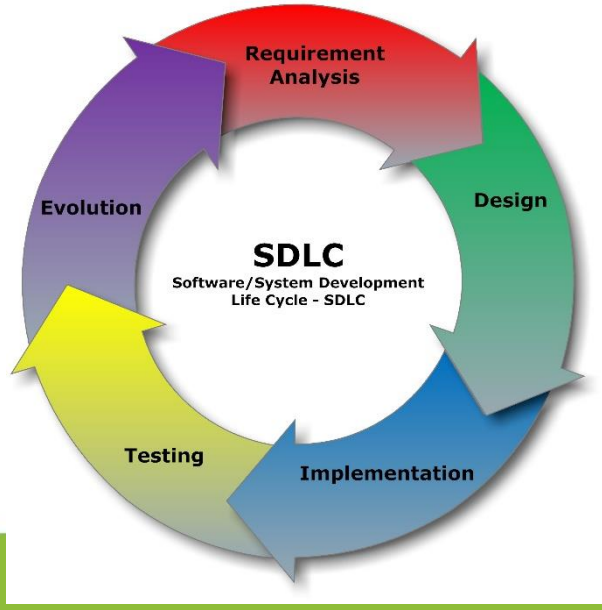
Inhoud

- Klassendiagram
- UML

Een klassendiagram

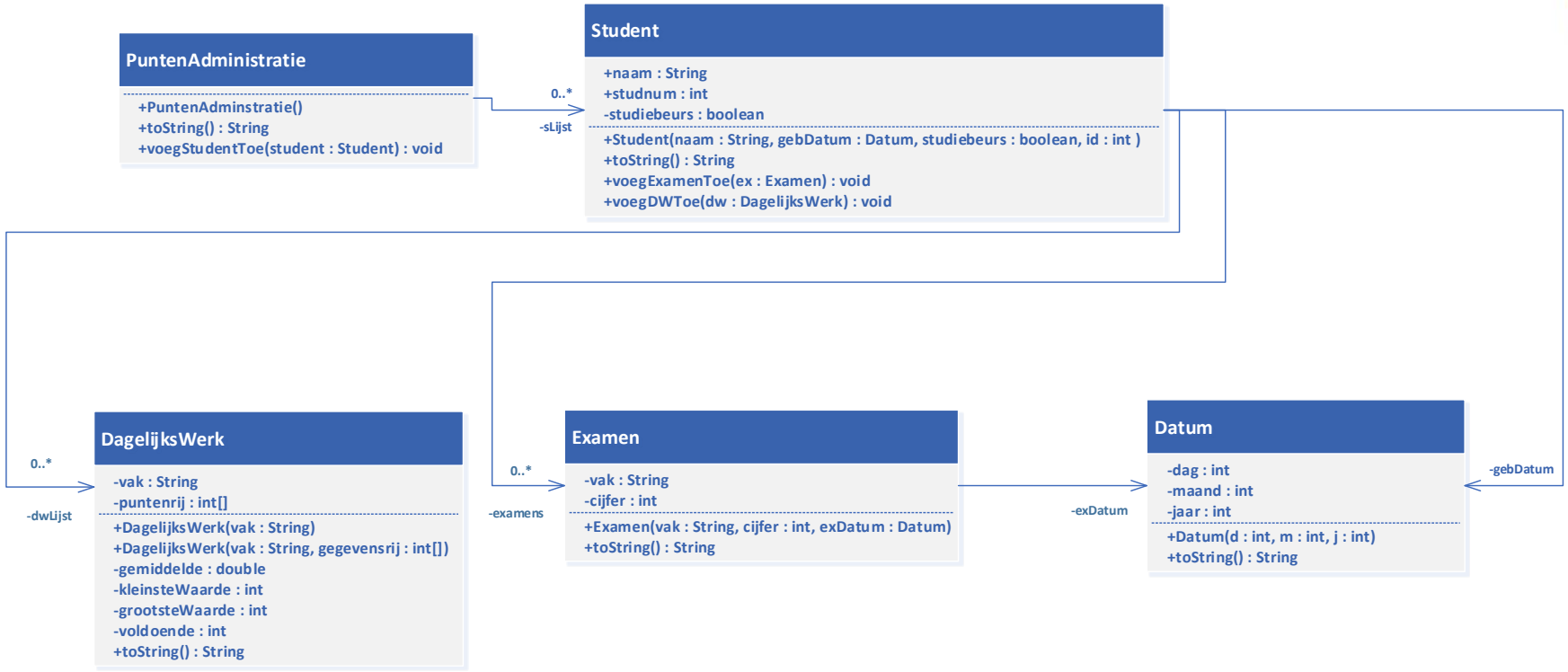
- **Een klassendiagram?**

  Een klassendiagram is een **schematische voorstelling** van één of meerdere klassen. Het wordt gebruikt om je probleem te analyseren en om tijdens de ontwerp-fase een oplossing te formuleren. In een later stadium tijdens de **SDLC** kan het gebruikt worden als ondersteunende documentatie.



Een klassendiagram

- **Voorbeeld: *puntenadministratie***



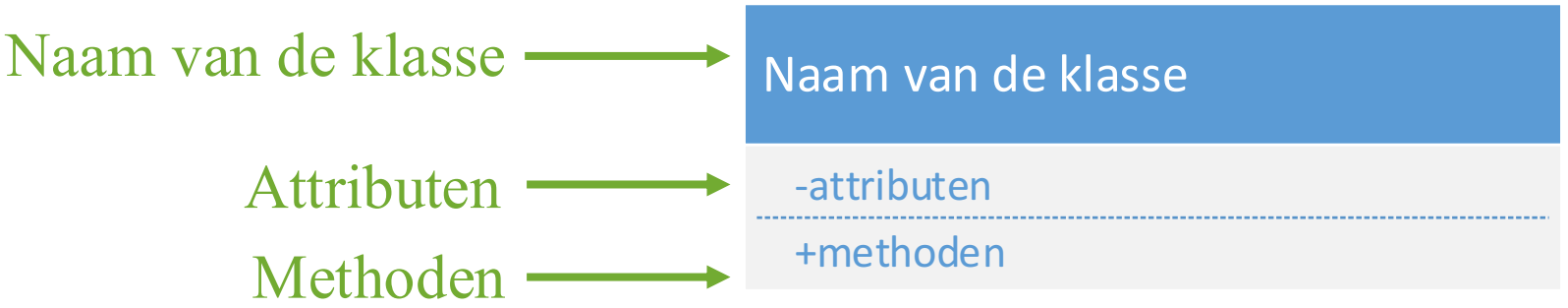
Een klassendiagram

- **Enkele voordelen**:
  - ✓ Het geeft een structureel overzicht van je programma.
  - ✓ Het geeft een overzicht van alle klassen en hun interactie.
  - ✓ Het geeft de mogelijkheid om verschillende programmeurs met elkaar te laten communiceren zonder dat er effectief over code moet gesproken worden.
  - ✓ Het wordt meestal voorgesteld door een gestandaardiseerde taal (bijv.: UML)
  - ✓ Het is herbruikbaar.
  - ✓ Het laat je toe om je oplossing visueel te evalueren.
  - ✓ ...

  Het heeft natuurlijk als voornaamste nadeel dat het tijd in beslag neemt om te maken.

UML

- Unified Modeling Language (**UML**) bestaat uit een aantal standaarddiagrammen die objectgeoriënteerde systemen grafisch kunnen voorstellen.



UML Diagram Toetsenbord

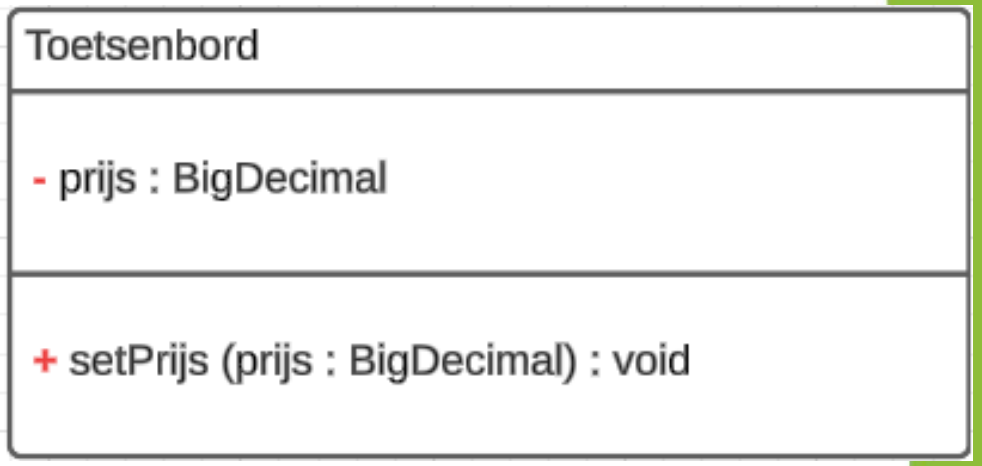
Toetsenbord klasse

UML Data Type en Parameter Notatie

- UML diagramma's zijn **onafhankelijk** van de programmeertaal
- UML diagrams gebruiken een onafhankelijke notatie om return types, **access modifiers**, datatypes, enz. te tonen.

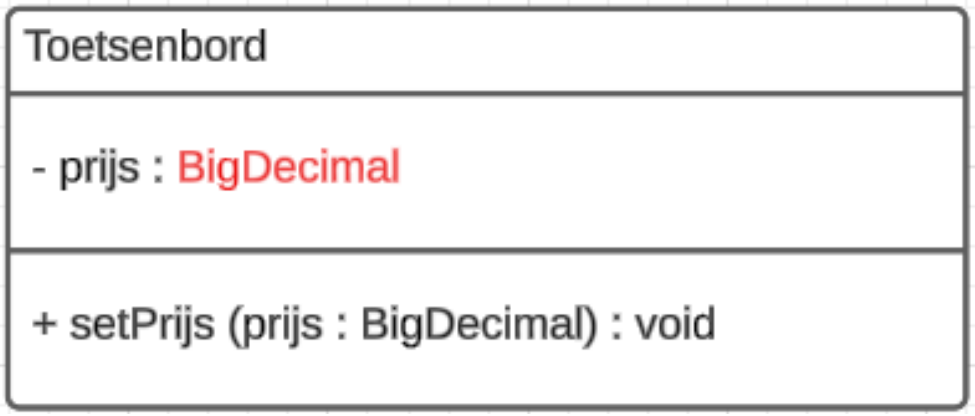
Access modifiers worden weergegeven als:

- \- private
- \+ public
- \# protected



UML Data Type en Parameter Notatie

- UML diagramma's zijn onafhankelijk van de programmeertaal
- UML diagrams gebruiken een onafhankelijke notatie om return types, access modifiers, **datatypes**, enz. te tonen.

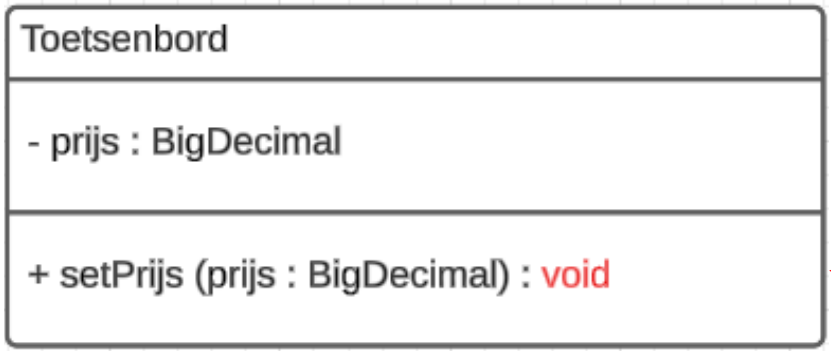


Datatypes van de variabelen worden achter de variabele naam geplaatst, gescheiden door een dubbelpunt.

UML Data Type en Parameter Notatie

- UML-diagramma's zijn onafhankelijk van de programmeertaal
- UML diagrams gebruiken een onafhankelijke notatie om **returntypes**, access modifiers, datatypes, enz. te tonen.

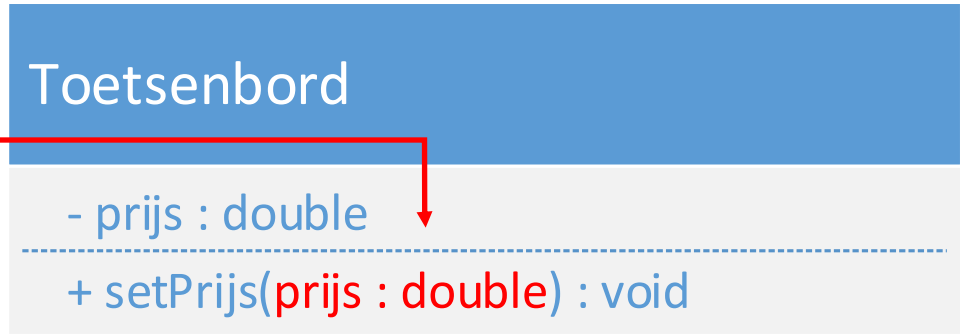
Returntypes van methoden worden na de methodenaam met parameters geplaatst, gescheiden door een dubbelpunt.



UML Data Type en Parameter Notatie

- UML diagramma's zijn onafhankelijk van de programmeertaal
- UML diagrams gebruiken een onafhankelijke notatie om return types, access modifiers, enz. te tonen.

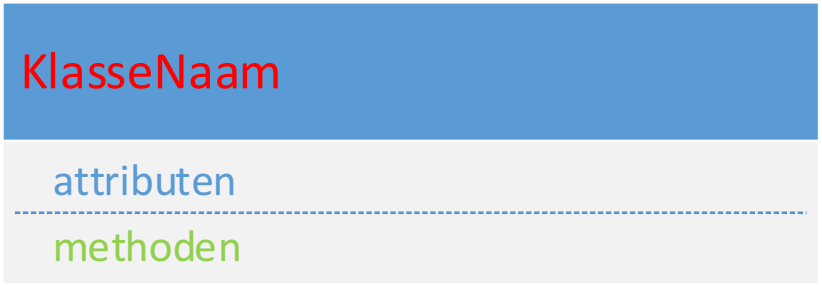
Methode parameters worden binnen de haakjes geplaatst volgens dezelfde notatie als de variabelen.



UML Diagramma converteren in Code

- Een Java class file kan dus eenvoudig omgezet worden naar een UML diagram.
- De compartimenten van een UML diagram komen overeen met de Java class file structuur.

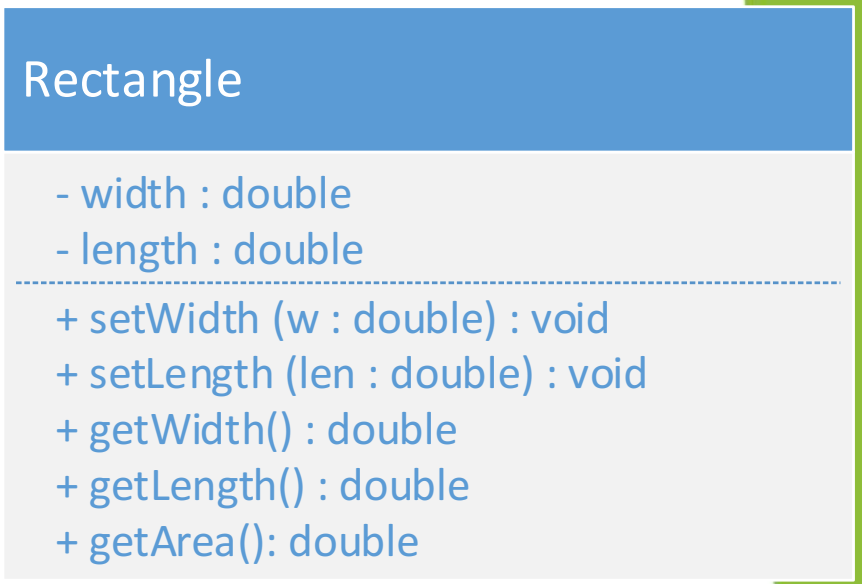
```
class KlasseNaam
{
   Attributen
   Methoden
}
```



Van UML Diagram naar Code

```
public class Rectangle
{
   private double width;
   private double length;

   public void setWidth(double w)
   {    width = w;
   }
   public void setLength(double len)
   {    length = len;
   }
   public double getWidth()
   {    return width;
   }
   public double getLength()
   {    return length;
   }
   public double getArea()
   {    return length * width;
   }
}
```



Constructors in UML

- In UML wordt een **constructor** als volgt voorgesteld: