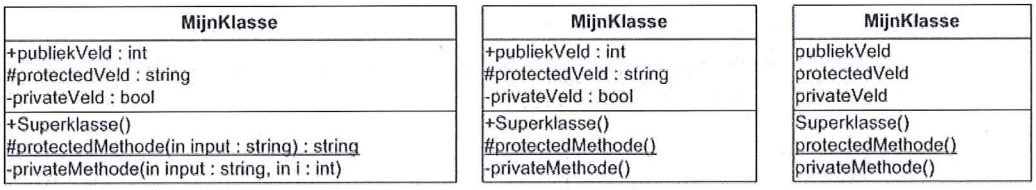
UML

UML staat voor Unified Modelling Language. UML doet dienst als werkinstrument in de ontwerpfase van een programmeerproject. Met de hulp van een UML ontwikkelomgeving wordt een applicatie 'ontworpen'. Dit ontwerp dient dan als basis voor de programmeur om de concrete implementatie te realiseren en te testen. Een aantal diagrammen kunnen worden opgesteld: UML Activity Diagrams, UML Collaboration Diagrams, UML Sequence Diagrams, ...

Deze cursus heeft niet als doel je deze ontwerptechnieken bij te brengen. Het klassendiagramma willen we je echter niet onthouden. De reden hiervoor is voor de hand liggend: voor een reeds geïmplementeerde toepassing is het heel eenvoudig een schematisch overzicht te bekomen van de gebruikte klassen, interfaces, ... Ook de onderlinge relaties en de member informatie kan overzichtelijk voorgesteld worden. Het is dé manier bij uitstek om overerving, compositie en implementatie van interfaces duidelijk te maken dankzij een grafische voorstelling.

Onderstaande UML schema's zijn manueel gecreëerd met behulp van de toepassing Microsoft Office Visio.

C.1 Klassen voorstellen



Figuur C.1: Verschillende schematische voorstellingen van een klasse.

Een klasse wordt voorgesteld door middel van een rechthoek waarin we drie

onderdelen onderscheiden:

- Bovenaan komt de klassenaam met eventueel de `package` waarin de klasse zich bevindt.
- In het midden vind je de velden met eventueel hun toegangsrechten.
- Onderaan komen alle methoden te staan: de constructoren, de statische en niet-statische methoden.

Toegangsrechten van velden en methoden kunnen aangegeven worden door vóór de signatuur hetvolgende te schrijven:

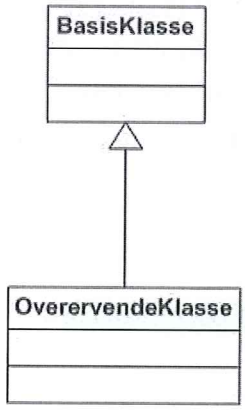
publiek een + teken

protected een # teken

private een -teken

Statische velden en methoden worden onderlijnd weergegeven. De parameters en het return type kan je al dan niet weergeven in het schema — selecteer in Visio de klasse, klik met de rechter muisknop en selecteer Shape Display Options.

C.2 Overerving

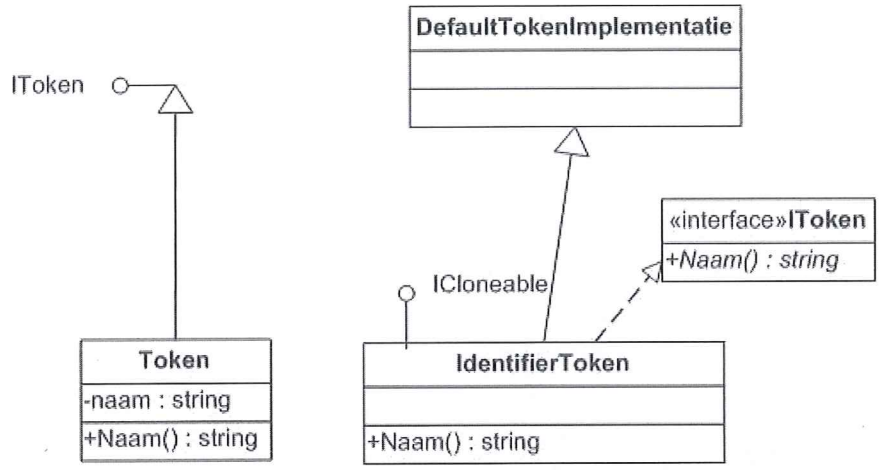


Figuur C.2: Schematische voorstelling van de overerving van klassen en de implementatie van interfaces.

Overerving tussen 2 klassen wordt voorgesteld door een pijl met de pijlpunt gericht naar de basisklasse.

Een interface kan worden voorgesteld als een cirkeltje waaraan een horizontaal streepje hangt (*lollipop shape*) met de naam van de interface erbij vermeld. Bij de

andere voorstelling van een interface bekom je een rechthoek met twee onderdelen (*class-like shape*): bovenaan komt de interface naam, eventueel voorafgegaan door de vermelding van het redundante `<<interface>>`. Onderaan komt de verzameling van methode definities van deze interface. Het overschakelen tussen de twee interface voorstellingen gebeurt door met de rechter muisknop op de interface te klikken en één van de volgende onderwerpen te selecteren: `Show as Lollipop Interface` of `Show as Class-like Interface`.

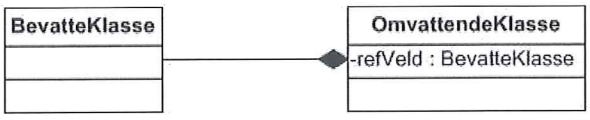


Figuur C.3: De klasse `Token` implementeert de `IToken` interface. De klasse `IdentifierToken` erft van de klasse `DefaultTokenImplementatie` en implementeert de interfaces `ICloneable` en `IToken`.

Als de relatie tussen een interface en zijn geïmplementerende klasse niet wordt weergegeven in je UML schema, dan kan je dit alsnog bekomen. Klik met de rechter muisknop op de interface en selecteer de optie `Show Relationships`.

C.3 Compositie

Een compositie relatie wordt voorgesteld door een verbinding met aan één van de uiteinden een gevulde ruit. De ruit staat bij de klasse die de andere klasse als veld omvat.



Figuur C.4: Schematische voorstelling van compositie.