Leerkracht: S. Vermeulen
Vak: Software Development
Klas: 6A&D

'Heeft een'-relaties (object als attribuut)

1. UML

a. Algemeen

- juiste pijl ⟶
- juiste richting van de pijl (je leest "heeft" in de richting van de pijl).
- multipliciteit vermelden minimum..maximum (eventueel *)
- relatienaam voorafgegaan door - (de relatienaam is de naam van de childklasse, maar dan met kleine letter).

*b. Specifiek voor multipliciteit minimum..**

- in de child-klasse:
  - een attribuut code voorzien (een unieke id)
- in de parent-klasse:
  - een voegToe-methode voorzien (bv. voegComputerToe)
  - een schrap-methode voorzien (bv. schrapComputer)
  - een opvraag-methode voorzien (bv. vraagComputerOp)
  - een methode voor het aantal children voorzien (bv. geefAantalComputers)

2. Code

a. Relatie met multipliciteit 0..1 of 1..1

- in de parent-klasse:
  - een attribuut voorzien met de naam van de childklasse, maar met kleine letter
  - getter en setter voorzien voor dit attribuut
  - Als multipliciteit 1..1, mag de setter van dit attribuut niet uitgevoerd worden als de parameter *null* is.
  - De toString van de parent-klasse moet ook de toString van de child-klasse uitvoeren.

b. Relatie met andere multipliciteit

- in de parent-klasse:
  - een attribuut voorzien dat een ArrayList is van de parent-klasse
  - in de constructor de new (instantiatie) van de ArrayList toepassen
  - voegToe-methode voorzien (eventueel checken op maximum van multipliciteit)
  - schrap-methode voorzien (eventueel checken op minimum van multipliciteit)
  - opvraag-methode voorzien
  - geefTotaalAantal-methode voorzien (die de grootte van de ArrayList teruggeeft)
  - toString moet de toString van de children oproepen