# Leerdoelen

## Assignment 1

Na afloop van deze opdracht kun je:

* je weg vinden in de programmeeromgeving Greenfoot;
* methodes aanroepen en hun effect analyseren;
* het verschil tussen een mutatormethode en een accessormethode in jouw eigen woorden omschrijven;
* aan de hand van een klassendiagram, de subklassen van een klasse benoemen;
* (geërfde) methodes van een klasse vinden en benoemen;
* eigenschappen van de toestand van een object benoemen;
* omschrijven wat de types int, String, void en boolean zijn;
* aan de hand van een signatuur, het resultaattype en de parameter(type)s van een methode benoemen;
* de relatie tussen een algoritme, een stroomdiagram en programmacode omschrijven;
* JavaDoc commentaar aan code toevoegen;
* stapsgewijs codeaanpassingen doorvoeren en testen;
* syntax foutmeldingen herkennen en interpreteren;
* afdrukken naar een console.

## Assignment 2

Na afloop van deze opdracht kun je:

* een algoritme ontwikkelen voor een probleem door de benodigde opeenvolgingen, beslissingen en herhalingen voor een oplossing te identificeren;
* aan de hand van begin- en eindsituaties, redeneren over de correctheid van een algoritme;
* een algoritme weergeven in een stroomdiagram;
* een stroomdiagram omzetten naar programmacode;
* sybmethodes combineren om problemen op te lossen;
* NIET, EN en OF en andere boolean methodes toepassen in een conditioneel statement;
* een return-statement gebruiken in een stroomdiagram en in code;
* gestructureerd en stapsgewijs code aanpassingen doorvoeren en testen;
* fouten opsporen in code, foutmeldingen analyseren en deze gebruiken om problemen op te lossen (debuggen);
* een generiek algoritme opstellen en implementeren.

## Assignment 3

Na afloop van deze opdracht kun je:

* een probleem opdelen in deelproblemen;
* decompositie toepassen in stroomdiagrammen en code door submethodes te herkennen;
* submethodes afzonderlijk ontwerpen, schrijven en testen;
* de rol van het resultaattype en de parameter(s) van een methode omschrijven;
* bestaande submethodes en oplossingen opnieuw gebruiken;
* conditionele expressies samenstellen door logische operatoren te combineren;
* nesting toepassen als strategie voor het oplossen van complexere problemen;
* generieke oplossingen ontwikkelen;
* jouw oplossing beoordelen.

## Assignment 4

Na afloop van deze opdracht kun je:

* uitleggen waarvoor variabelen gebruikt worden;
* een variabele declareren, initialiseren en een waarde toekennen;
* code en de daarin gebruikte variabelen traceren met behulp van een tracetabel;
* de (waarde)toekenningsoperator ‘=’ gebruiken;
* de vergelijkingsoperatoren ‘==’, ‘!=’, ‘<’, ‘<=’, ‘>’ en ‘>=’ gebruiken;
* de rekenkundigeoperatoren ‘+’, ‘-’, ‘\*’, ‘/’ en ‘%’ gebruiken;
* de verhogingsoperator ‘++’ (en verlagingsoperator ‘--’) gebruiken;
* plannen voor tellen, herhalingen en voor het verwisselen van waarden herkennen en toepassen;
* een teller gebruiken om te bepalen hoe vaak bepaalde code herhaald moet worden (plan: teller-gestuurde herhaling);
* een conditie gebruiken om steeds te bepalen of bepaalde code herhaald moet worden (plan: conditie-gestuurde herhaling);
* de rol van een parameter (van een methode of constructor) uitleggen;
* controleren op ongeldige waarden.

## Assignment 5

Na afloop van deze opdracht kun je:

* standaard plannen herkennen en toepassen: voor een verzameling elementen het maximum/minimum bepalen en de som en het gemiddelde berekenen;
* type-casting toepassen;
* een generiek algoritme ontwikkelen en implementeren;
* een probleem opdelen in deelproblemen en die afzonderlijk oplossen.

## Assignment 6

Na afloop van deze opdracht kun je:

* List-variabelen declareren en gebruiken;
* Java Library Documentation gebruiken om bestaande Java methodes op te zoeken;
* bestaande methodes van een List toepassen, zoals het bekijken en verwijderen van elementen;
* een for-each-loop gebruiken voor het doorlopen van de elementen van een lijst;
* elementen in een lijst met elkaar verwisselen;
* een lijst sorteren;
* een lijst gebruiken om primitieve typen (zoals integers of strings) of objecttypes (zoals Egg) op te slaan;
* het plan voor het uitrekenen van de gemiddelde toepassen op een lijst.

## Assignment 7

Na afloop van deze opdracht kun je:

* random (willekeurige) getallen gebruiken om onvoorspelbaarheid toe te voegen;
* klasseconstantes (globale variabelen waarvan de waarde niet kan veranderen) gebruiken;
* een complex algoritme op splitsen in deeltaken en daarna implementeren.