**Oefening hoofdstuk 23 Interfaces**

Prey Predator

# Prey predator

We maken een simulatie van roofdieren en hun prooi.

In deze simulatie zijn de prooidieren luizen en de roofdieren lieveheersbeestjes.

We starten met de implementatie van een abstracte klasse Insect. Ieder insect heeft de volgende eigenschappen:

* XCoord en YCoord (de positie van het insect in zijn leefwereld, je mag hiervoor ook gebruik maken van een eigen klasse Position). De x- en y-coördinaat moeten steeds een waarde hebben van 0 tot 15 (incl.).
* Age: leeftijd, is het aantal ronden in de simulatie dat het insect heeft overleefd.
* MoveForward, MoveBack, MoveLeft, MoveRight: verander de positie van het insect in zijn leefwereld. MoveForward en MoveBack verhogen of verlagen de YCoord, MoveLeft en MoveRight verhogen of verlagen de XCoord. Indien je gebruik maakt van een klasse Position, kan je deze methodes hier voorzien.

Voorzie nu de volgende interfaces:

1. IPrey

Deze interface bevat:

* 1. Een read-only property voor de x-coordinaat en y-coordinaat van een prooidier
  2. Een methode Breed() die als resultaat een nieuw IPrey object geeft.

1. IPredator

Deze interface bevat:

* 1. De methode Chase() waarbij je lijst van prooidieren meegeeft en de resultaatwaarde is opnieuw een lijst van prooidieren
  2. De methode Eat() waarbij je een lijst van prooidieren meegeeft als parameter.
  3. De methode Distance(IPrey) waarmee de afstand tussen de jager en het prooidier berekend kan worden.
  4. De methode Breed() die als resultaat een nieuw IPredator object geeft
  5. De read-only property Starving

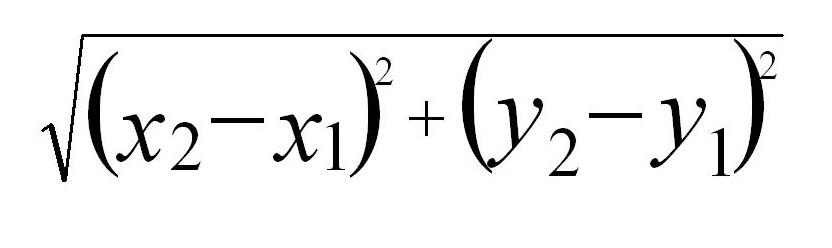
We maken nu de klasse Louse (luis) en de klasse Ladybird (of Ladybug) (lieveheersbeestje).

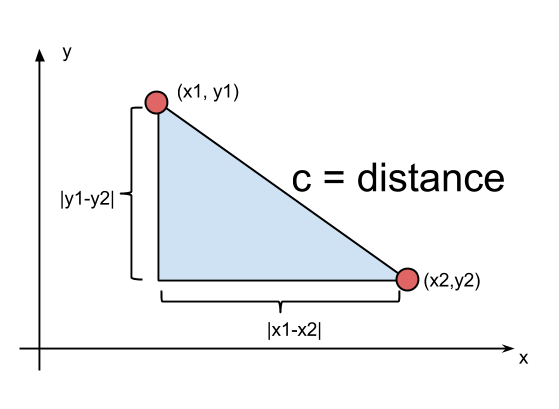
Voor de klasse Louse, die de interface IPrey implementeert, voorzie je het volgende gedrag:

* Breed: na 5 ronden in de simulatie maakt een luis een nieuw luis-object aan. Het nieuwe luis-object heeft dezelfde coördinaten als zijn ouder. Na het produceren van een nakomeling zal het terug 5 ronden duren vooraleer de luis een nieuwe nakomeling kan produceren.

Voor de klasse Ladybird, die de interface IPredator implementeert, voorzie je het volgende gedrag:

* Breed(): Als een lieveheersbeestje 6 ronden kan overleven, zal er een nakomeling aangemaakt worden (op dezelfde manier als de luis).
* Chase(): Een lieveheersbeestje jaagt op een lijst van prooien. Alle prooien die op een afstand met waarde 2 (of minder) van het lieveheersbeestje afstaan, worden gevangen en dus teruggegeven als resultaat. Het lieveheersbeestje kan maximaal 20 prooien vangen!
* Distance(IPrey): bereken de afstand tot een prooi. Gebruik hiervoor de volgende formule:





In deze formule zijn (x1, y1) de coördinaten van het roofdier en (x2, y2) de coördinaten van de prooi.

* Eat(): Het lieveheersbeestje eet zijn gevangen prooien. Je moet hier enkel controleren dat het lieveheersbeestje een prooi heeft kunnen bemachtigen (en dus deze ronde niet zal sterven).
* Starving: Deze eigenschap geeft true, wanneer het lieveheersbeestje gedurende 3 ronden geen prooien heeft kunnen eten.

Bij het starten van het programma maak je nu een lijst met 4 roofdieren en een tweede lijst met 100 prooidieren. Voor de x- en y-coördinaat van ieder insect genereer je een willekeurig getal tussen 0 en 15 (incl.).

In de window voorzie je labels om het aantal roofdieren en het aantal prooidieren te tonen. Je voorziet verder een knop. Een klik op deze knop simuleert 1 ronde.

In iedere ronde wordt de volgende instructies uitgevoerd:

1. Alle roofdieren en prooidieren bewegen willekeurig (MoveForward, MoveBack, MoveLeft of MoveRight) en worden een jaartje ouder.
2. Vervolgens gaat ieder roofdier op jacht (Chase) en eet hij de gevangen prooien (Eat). Zorg ervoor dat de gevangen prooidieren correct verwijderd worden uit de lijst van prooidieren.
3. Vervolgens controleer je of het roofdier stervende is (Starving). Indien dit het geval is, verwijder je het roofdier uit de lijst van roofdieren.
4. Tenslotte gaan er nakomelingen geproduceerd worden, zowel voor prooi- als roofdieren, en toon je de aantallen van beide groepen in de user interface.

We gaan onze prooien en roofdieren ook weergeven in een canvas.

Hiervoor maken we eerst een interface IDisplayable. De klasse Insect zal deze interface implementeren.

De interface voorziet de volgende methoden:

* void DisplayOn(Canvas canvas, Color color)
* void RemoveFromCanvas()
* void UpdateDisplay()

In de abstracte klasse Insect zal je in de implementatie van DisplayOn(…) ervoor zorgen dat een ellipse met de opgegeven kleur in het canvas getoond zal worden.

Bij het verplaatsen van een insect zal je UpdateDisplay() aanroepen, waarbij je de plaats van de ellipse in het canvas aanpast.

Zorg dat in het hoofdprogramma voor alle luizen en lieveheersbeestjes op het juiste ogenblik de DisplayOn(…) en RemoveFromCanvas() wordt aangeroepen.

Tip: je kan ook IPrey en IPredator de interface IDisplayable laten overerven.

