



Opleiding: Java EE

Resultaat 44.50/

Naam test: fundamentals

Datum: 26/04/21

Lesmodules:

Naam cursist: Bifal Abz

TEST: FUNDAMENTALS

FORESTBOOK: NOTEBOOK VOOR DE WAKKERE WANDELAAR

/50

In deze test gaan we een applicatie schrijven die het mogelijk maakt aan een gebruiker om de observaties die deze maakt in het bos, bij te houden in een notebook.

De applicatie zal stap voor stap uitgelegd worden, indien je alsnog vragen hebt of er zaken niet duidelijk zijn kan je deze vragen aan de instructeur stellen.

De meeste functionaliteit uit dit project zal zich in de App klasse bevinden, alsook in de klasse ForestBook:

- Planten toevoegen
- Dieren toevoegen
- Planten en dieren sorteren
- dieren filteren in een lijst[Carnivore, Herbivore, Omnivore]
- dieren en planten printen

VERGEET ZEKER DE GOOD PRACTICES NIET[DEZE STAAN BESCHREVEN IN HET DOCC]

Veel succes!

1.1 ENTITIES

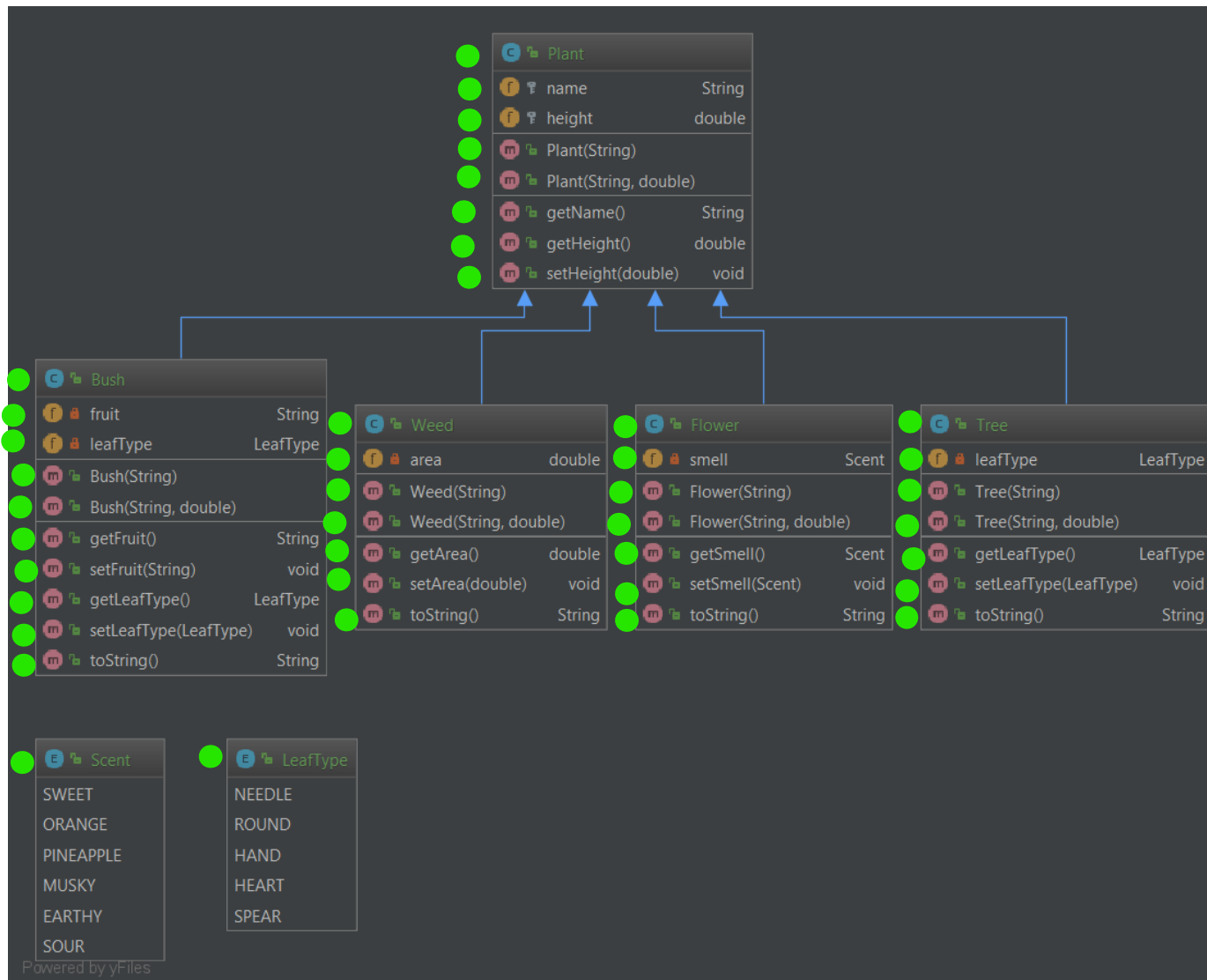
- O

Je begint door in de src folder een nieuwe package aan te maken noem deze package: (be.intecbrussel.the_notebook.)entities

- O 1.2 PLANT_ENTITIES

In de package entities maak je een nieuwe package: plant_entities

- O Maak de volgende klassen in deze package [neem UML aandachtig over]



lets meer info over deze klassen

Scent en LeafType zijn enums, deze worden in sommige subclasses gebruikt

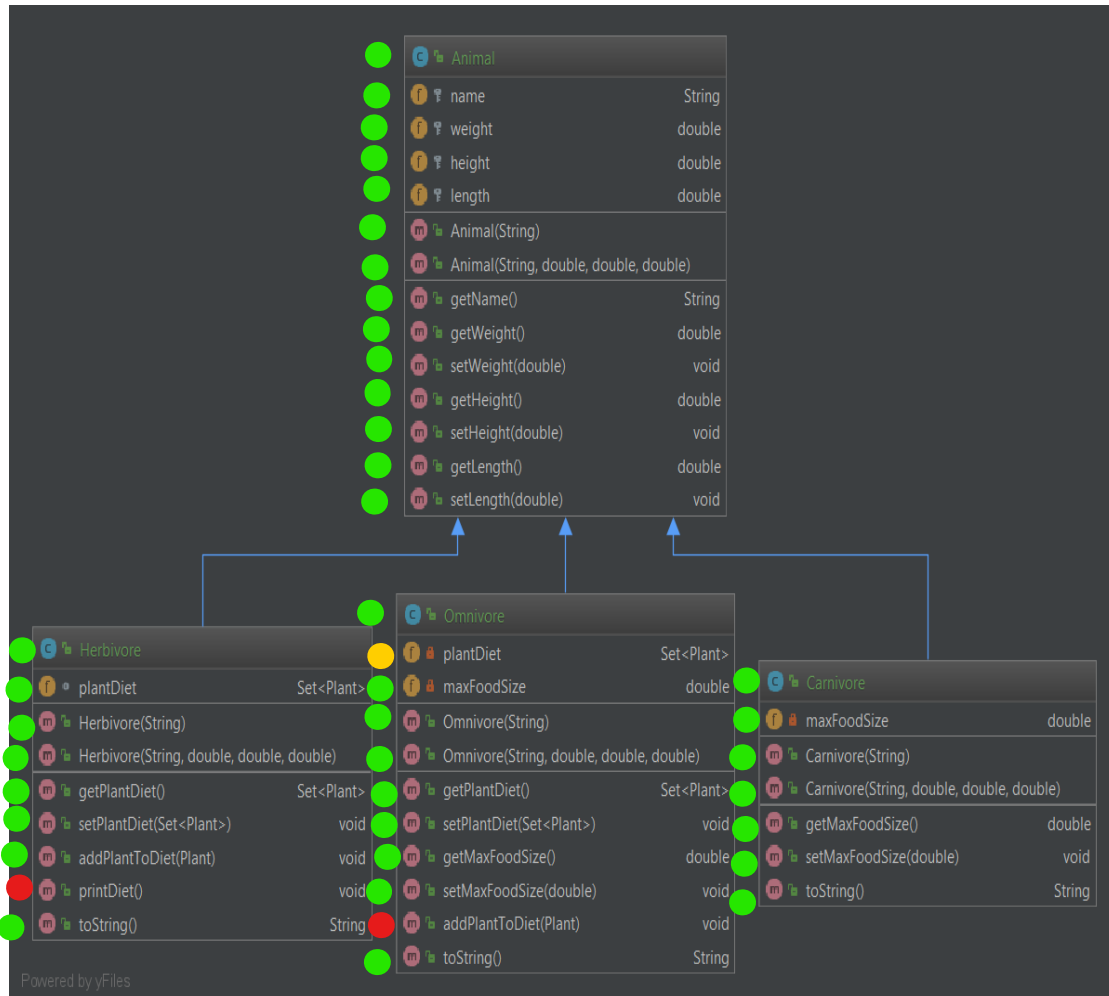
De bovenstaande planten zijn de verschillende soorten planten die een onderzoeker kan tegenkomen tijdens een wandeling in het bos.

De eigenschap name zal telkens de naam van het soort zijn vb "Pink Delight"
[wordt meegegeven aan de constructor tijdens initialisatie in de App klasse].

Verder is er ook de mogelijkheid om andere eigenschappen[zoals Scent of LeafType] in te stellen in de Appklasse via setters[zie uml].

1.3 ANIMAL_ENTITIES

- O _____
maak op dezelfde hoogte als plant_entities een nieuwe package aan : animal_entities
- O Maak de volgende klassen in deze package [neem UML aandachtig over]



Let op! De name van de Animal klasse heeft geen setter methode.

Carnivore houdt een `maxFoodSize` bij. Hierin gaan we bijhouden wat de maximum hoogte van de prooi is, dat dit dier kan opeten.

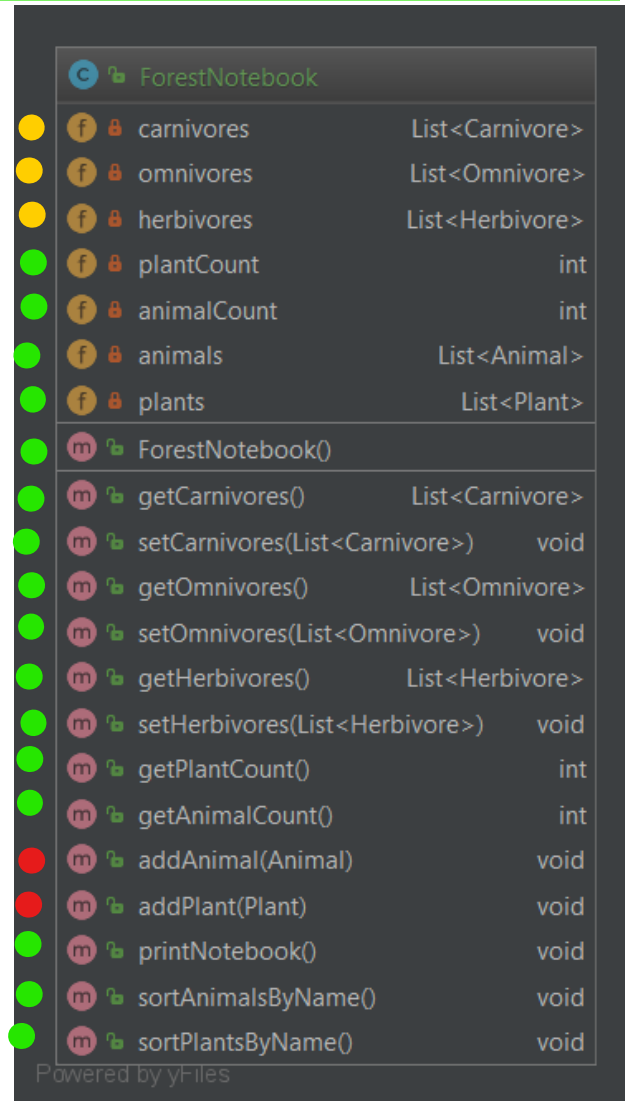
Herbivore maakt gebruik van een set waar we verschillende planten in kunnen opslaan. Deze planten stellen de planten voor die een herbivore opeet.

Met de `addPlantToDiet` methode kan een gebruiker een plant toevoegen, we zullen deze intern bijhouden in onze eigenschap `plantDiet`.

Omnivore is een combinatie van een carnivore en een herbivore. [Hierin houden we dus zowel een `maxFoodSize` als `plantDiet` bij, voor implementatie zie bovenstaande uitleg]

1.4 SERVICE

- O Maak een nieuwe package aan die zich op dezelfde hoogte als de entities package bevindt en noem deze package: service
- O Maak nu onderstaande klasse [volg het uml schema aandachtig]



Bovenstaande klasse stelt een digitaal notitieboek voor waar we alle dieren en planten in kunnen opslaan.

De methodes om planten en dieren toe te voegen moeten een controlemechanisme hebben. Ze moeten controleren of er niet 2 keer hetzelfde dier wordt toegevoegd. **[vergelijk eventueel de naam]**

Verdere implementatie details:

- ☐ de plantCount methode retournt de hoeveelheid planten die we intern bijhouden
- ☐ de animalCount methode retournt de hoeveelheid dieren die we intern bijhouden
- ☐ de printNotebook methode print alle planten en dieren. **[dit kan je op verschillende manieren doen, probeer zelf de beste strategie te kiezen]**
- ☐ de sortAnimalsByName methode sorteert de interne animal lijst op naam.
- ☐ de sortPlantsByName methode sorteert de interne plantenlijst op naam.

TIP: Je krijgt in de methode addAnimal een argument Animal animal mee deze zal zowel in de animals list als in de juiste interne list moeten worden toegevoegd **[dus in carnivores, herbivores of omnivores]**.

TIP: Je mag zelf kiezen welke concrete implementatie je voorziet voor de list **[lees: ArrayList of LinkedList]** bij de verschillende eigenschappen **carnivores, herbivores, omnivores, plants en animals**.

BONUS [enkel indien tijd over]

- ☐ maak een sortAnimalsByHeight methode en een sortPlantsByHeight methode.

1.5 APP

- O Maak een nieuwe package aan die zich op dezelfde hoogte als de entities en service package bevindt en noem deze package: app
- O Maak nu in deze package een klasse NatureApp aan **[deze bevat uiteraard een main methode]**.

Deel 1 info

De bedoeling is dat je de rol van een boswandelaar voor je rekening neemt. Tijdens je ontspannende zondagochtend wandeling, kom je oog in oog te staan met een aantal dieren en planten. Nauwlettend meet je alle eigenschappen van de dieren en planten. En hou je deze dieren en planten bij in je notebook.

- O Maak een instantie aan van Notebook en geef deze een zinnige naam.
- O Maak minstens 5 verschillende planten aan, zorg ervoor dat je zeker 1 Tree, Flower, Weed en Bush object hebt. **[gebruik hiervoor de juiste constructor, alsook methoden om alle metingen bij te houden]**
- O Maak nu minstens 9 verschillende dieren aan. Let er op dat je minstens 1 Herbivore, Carnivore en Omnivore object hebt. **[gebruik hiervoor de juiste constructor, alsook methoden om alle metingen bij te houden]**

TIP: Vergeet niet dat sommige dieren planten eten en je hiervoor een plantDiet meegeeft aan deze objecten **[Je kan gerust dezelfde Set meermaals aan de verschillende instanties geven]**.

Deel 2 info

Na de wandeling kom je terug thuis en blader je even door je notebook om je nieuwe ontdekkingen te zien.

- ☐ gebruik een `sysout` om het aantal dieren en planten die je bent tegengekomen te tonen[**hint: zou minimum 5 planten en 9 dieren moeten zijn**].
- ☐ Gebruik vervolgens de methode `printNotebook` van je Notebook instantie[**toont de ongesorteerde lijst van animals en plants**].
- ☐ Print de lijsten van **carnivores**, **omnivores** en **herbivores** afzonderlijk af.
- ☐ Sorteer de dieren en planten met de `sortAnimalsByName` en `sortPlantsByName` methoden van je Notebook instantie.
- ☐ Gebruik vervolgens de methode `printNotebook` van je Notebook instantie[**toont nu de gesorteerde lijst van animals en plants**].

TIP: indien je beslist om **private helper methoden** te maken in **ForestBook** of **NatureApp** probeer deze dan te voorzien van commentaar die een beknopte omschrijving van de methode geeft.