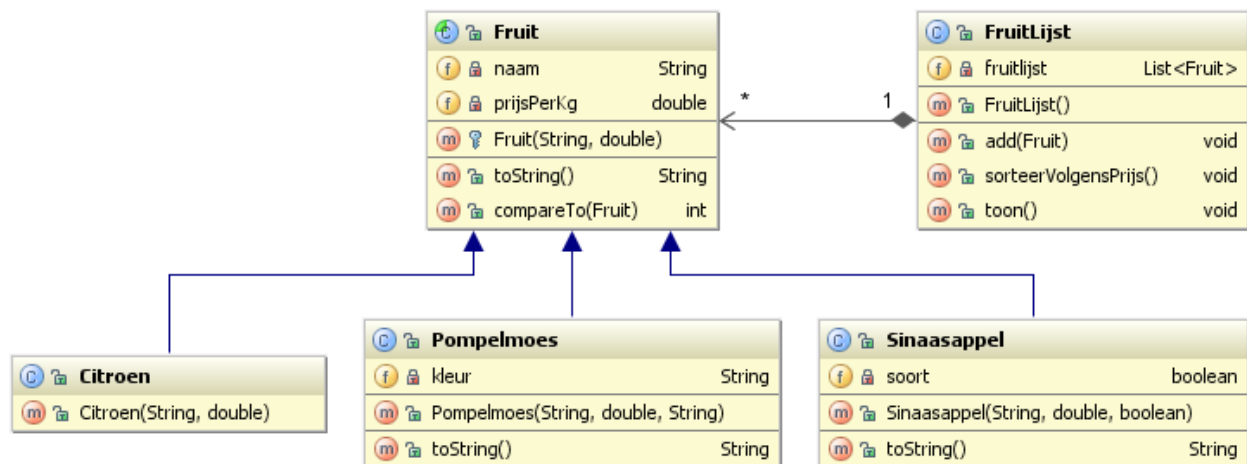


HERHALINGSOEFENINGEN: MAVEN

1 FRUIT



Maven INFO:

- Zorg ervoor dat je in dit project kan gebruik maken van de nieuwigheden in JAVA 8.
- De standaard source-encoding mag staan op UTF-8.
- Voeg een dependency toe om te kunnen werken met **slf4j**. Maak gebruik van de **log4j2** implementatie van **slf4j**.
SLF4J= Logging Framework.
Documentatie: <https://www.slf4j.org/apidocs/index.html>
- Voeg een property toe om het log level in te stellen op WARN. Gebruik deze property in je property file of xml file van Log4J.
- Voeg een maven profile toe genaamd: DEV. Hier zet je de property van het log level op DEBUG.

Project INFO:

De klasse Fruit

De klasse Fruit is een **abstracte klasse** die tevens de **Comparable interface** implementeert. Deze interface is een bestaande interface in Java, je hoeft hem dus niet zelf te schrijven.

Werk de methode `compareTo(Fruit fruit)` zo uit dat de `prijsPerKg` van de objecten met elkaar vergeleken wordt.

De klasse Citroen

Deze klasse is een uitbreiding van de klasse Fruit. Je hoeft alleen een constructor voorzien voor deze klasse.

De klasse Pompelmoes

De klasse Pompelmoes is een uitbreiding van de klasse Fruit. Ze heeft een extra variabele kleur van het type String om de kleur van de pompelmoessoort aan te duiden.

Hier moet je naast een constructor ook een toString methode voorzien.

De klasse SinaasAppel

De klasse SinaasAppel is een uitbreiding van de klasse Fruit. Ze heeft een extra variabele van het type boolean om aan te duiden of het om een eet- of een persappelsien gaat (true bij eetappelsien). Hier moet je naast een constructor ook een toString methode voorzien.

De klasse FruitLijst

De klasse FruitLijst bevat slechts één variabele, met name een ArrayList van Fruit- objecten.

- ✓ Voorzie een lege constructor om de ArrayList fruitlijst te maken.
- ✓ Zorg voor een add methode om een Fruit object aan de List toe te voegen.
- ✓ Voorzie een methode sorteerVolgensPrijs waarbij je gebruik maakt van de methode Collections.sort().
- ✓ Zorg voor een methode toon waarmee je alle objecten in fruitlijst afdrukt.

SLF4J:

- Telkens er een fruitlijst wordt aangemaakt geef je informatie logging mee.
- Bovenaan elke methode geef je debug logging mee. In welke methode bevinden we ons momenteel?
- In de methode met parameter (add) voeg je extra debug logging toe, waar je gebruikt maakt van een placeholder om het argument ook mee uit te printen.

De klasse FruitApp

Hier mag je niets aan wijzigen. Deze klasse dient alleen om de goede werking van je programma te testen.

```
public class FruitApp {  
  
    public static Fruit[] fruitsoorten = new Fruit[] {  
        new Citroen("Algarve citroen", 3.0), new Sinaasappel("Jaffa appelsien", 3.2, true),  
        new Sinaasappel("Bloedappelsien", 3.4, true),  
        new Pompelmoes("Roze pompelmoes", 3.1, "roze"),  
        new Sinaasappel("Persappelsien", 2.5, false),  
        new Pompelmoes("Rode pompelmoes", 3.3, "rood"),  
        new Pompelmoes("Gele pompelmoes", 2.9, "wit"), };  
  
    public static void main(String[] args) {  
        FruitLijst fruitLijst = new FruitLijst();  
  
        for (Fruit fruit : fruitsoorten) {  
            fruitLijst.add(fruit);  
        }  
        fruitLijst.sorteerVolgensPrijs();  
        fruitLijst.toon();  
    }  
}
```

Doe een Maven clean install en voer je project uit. Verander het profile naar DEV en run je programma opnieuw. Normaal zou je nu alle logs moeten kunnen bekijken.