



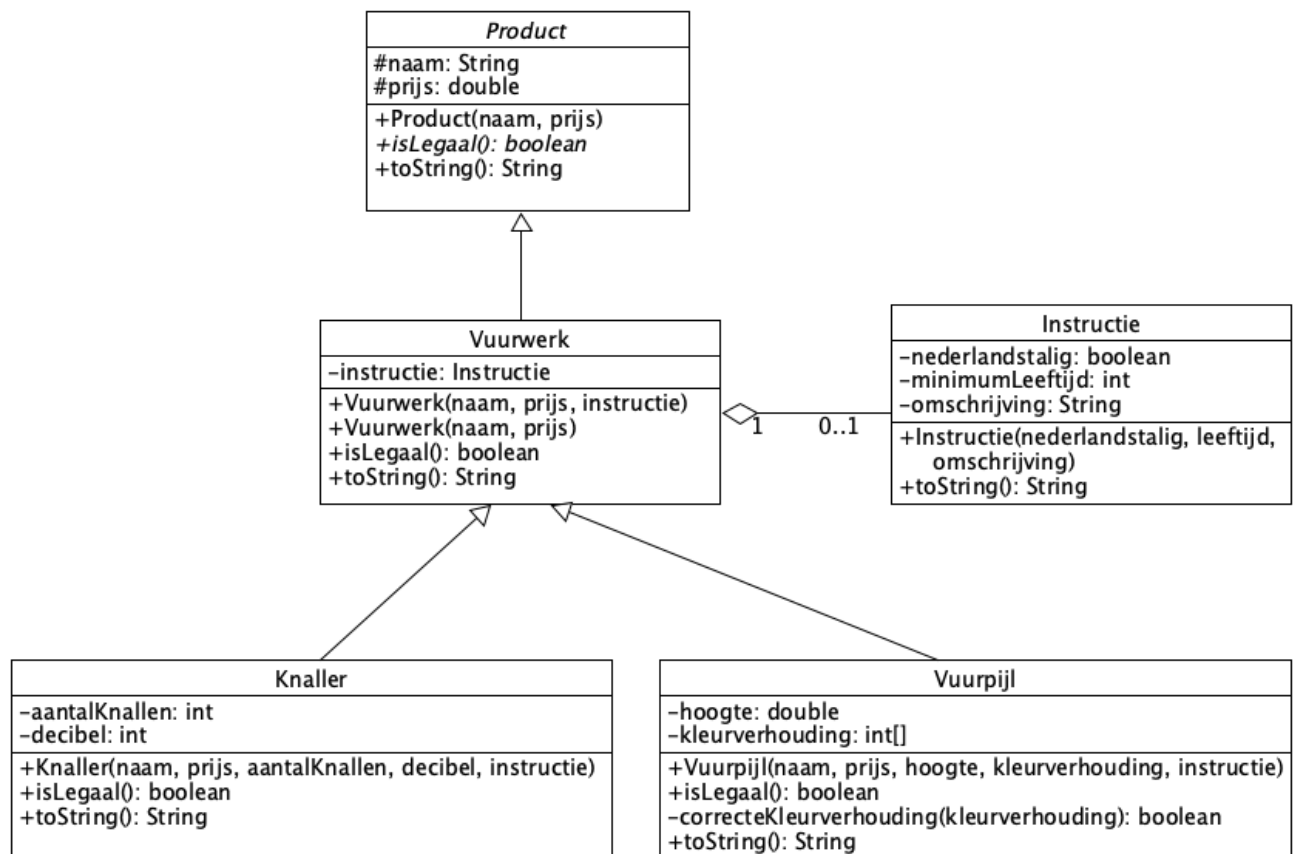
Happy New Year!

Instructies

- Je moet voor deze opgave code toevoegen aan een bestaand startproject. Download de zip van het startproject van OOP1 Moodle > Oefentoets > IntelliJ project.
- Maak na afloop een zipfile van je hele IntelliJ project en geef deze file als naam je voornaam + studentnummer aan elkaar, dus bijvoorbeeld: *Marie500812345*. Upload de zipfile op Moodle bij Oefentoets > Upload uitwerking.

Inleiding

De opdracht bestaat uit een aantal stappen. Om deze stappen te kunnen testen is al veel testcode gemaakt in de class `HappyNewYearDriver` (niet weergegeven in het class diagram). Maak hiervan gebruik! De voorbeeldoutput is ook gebaseerd op deze testcode.



Bovenstaand class diagram moet worden geïmplementeerd. Bekijk het goed, en lees alle opmerkingen hieronder goed door!



Let op:

- Er zijn geen getters en setters getoond. Voeg zelf getters en setters toe die je denkt nodig te hebben. Meer mag, maar hoeft niet.
- Zorg dat je `toString()` methodes soortgelijke output genereren als in de outputvoorbeelden.
- De classes `Instructie` en `Product` zijn aanwezig, maar nog incompleet. Deze zal je moeten aanpassen (zie stap 1a en 1b).
- De subclasses `Vuurwerk`, `Knaller` en `Vuurpijl` zijn niet aanwezig. Deze zal je in het geheel moeten maken (zie stap 1c, 2 en 3).
- Je hoeft alleen Javadoc te schrijven voor de class `Instructie`. Voor de andere classes hoef je geen Javadoc te schrijven.
- Als je zelf nog zaken wil toevoegen die niet in het class diagram staan (omdat je denkt dat ze nodig zijn), geef dat dan duidelijk met commentaar aan in de code.

Deel 1: bouw klassenstructuur

Run de applicatie. De output moet er als volgt uit zien.

```
Happy new year from: <naam>, <studentnummer>, <klas>
```

Stap 0: Verander de nieuwjaarswens

Verander de nieuwjaarsgroet zodat je eigen naam, je studentnummer en je klas verschijnt.

Stap 1a: Class `Instructie` (10 pt)

De class `Instructie` is vrijwel geheel af. Je moet alleen twee dingen toevoegen:

- Voeg visibilij modifiers toe, zodat de regels voor data encapsulation correct zijn toegepast.
- Voeg Javadoc toe.

Stap 1b: Class `Product` (5 pt)

Maak de abstract class `Product` af volgens het class diagram. Let erop dat de methode `isLegaal()` abstract is.

Stap 1c Class `Vuurwerk` (15 pt)

Maak de class `Vuurwerk`. Houd hierbij rekening met de volgende eisen:

- De class heeft twee constructors. Implementeer de tweede constructor door de eerste aan te roepen (constructor chaining).
- De methode `isLegaal()` geeft aan of het vuurwerk legaal is. Voor al het vuurwerk in deze applicatie geldt: vuurwerk is legaal als het een instructie heeft, en als deze instructie Nederlandstalig is.
- Voor de methode `toString()` geldt:
 - Alle attributen moeten op een nieuwe regel staan.



- Alle attributen van de superclass moeten ook verschijnen. Dit laat dus ook de naam en de prijs zien.
- Voor de prijs moet het euro-teken staan. Gebruik hiervoor indien nodig “\u20AC”.
- Indien er geen Instructie is moet er “ontbreekt” staan.
- Laat ook zien of het vuurwerk legaal of illegaal is.
- Test je code van stap 1a en 1b door in de Driver class alle code met TODO stap 1 te “uncommenten”.

De output zou er als volgt uit moeten zien.

```
--- STAP 1 ---
Naam: Veiligheidsbril
    Prijs: €2,50
    Instructie: Nederlandstalig=true, leeftijd=6, omschrijving=Draag bij aansteken
    Legaal: true
Naam: Safety glass
    Prijs: €2,50
    Instructie: Nederlandstalig=false, leeftijd=6, omschrijving=Wear before ignition
    Legaal: false
Naam: Aansteeklont zonder instructie
    Prijs: €0,25
    Instructie: ontbreekt
    Legaal: false
```

Stap 2: Subclass Knaller (15 pt)

Maak de subclass `Knaller`. Houd hierbij rekening met de volgende eisen:

- Override de methode `isLegaal()`. Een extra eis voor een `Knaller` is dat deze legaal is als het aantal decibel niet hoger is dan 120.
- Voor de methode `toString()` geldt:
 - Alle attributen moeten op een nieuwe regel staan.
 - Alle attributen van de superclass moeten ook verschijnen. Dit laat dus ook zien of het vuurwerk legaal of illegaal is.
- Test je code door in de Driver class alle code met TODO stap 2 te “uncommenten”.

De output zou er als volgt uit moeten zien.

```
--- STAP 2 ---
Naam: Celebration Crackers
    Prijs: €10.00
    Instructie: Nederlandstalig=false, leeftijd=21, omschrijving=Keep minimum 10 ft distance
    Legaal: false
    Aantal knallen: 777
    Decibel: 75
Naam: Peking Rol
    Prijs: €45.00
    Instructie: Nederlandstalig=true, leeftijd=21, omschrijving=Houd minimaal 5 meter afstand
    Legaal: true
    Aantal knallen: 500
```



```
Decibel: 120
Naam: Shanghai Rol
Prijs: €85.00
Instructie: Nederlandstalig=true, leeftijd=21, omschrijving=Houd minimaal 5 meter afstand
Legaal: false
Aantal knallen: 1000
Decibel: 125
Naam: Hongkong Rol
Prijs: €82.50
Instructie: Ontbreekt
Legaal: false
Aantal knallen: 1000
Decibel: 100
```

Stap 3: Subclass Vuurpijl (15 pt)

Codeer de subclass `Vuurpijl`. Houd hierbij rekening met de volgende eisen:

- Het attribuut "hoogte" geeft in meters weer hoe hoog de pijl de lucht in schiet.
- Het attribuut "kleurverhouding" is een array waarin van de 3 kleuren rood, groen en blauw de percentages staan. Zo is de kleur geel gecodeerd met het volgende array `[50, 50, 0]`, dat wil zeggen 50% rood, 50% groen en 0% blauw. Dit werkt natuurlijk alleen maar wanneer de som van de percentages 100 is.
De methode `correcteKleurverhouding(kleurverhouding)` controleert of de percentages samen 100 zijn. Gebruik deze methode in je constructor. Als de aan de constructor meegegeven kleur niet correct is, moet een foutmelding worden afgedrukt, en wordt de kleur op rood gezet `[100, 0, 0]`.
- Override de methode `isLegaal()`. Een extra eis voor een `Vuurpijl` is dat deze legaal is wanneer de instructie aangeeft dat de leeftijd minimaal 16 jaar is.
- Voor de methode `toString()` geldt:
 - Alle attributen moeten op een nieuwe regel staan.
 - Alle attributen van de superclass moeten ook verschijnen. Dit laat dus ook zien of het vuurwerk legaal of illegaal is.
- Test je code door in de Driver class de code alle code met TODO stap 3 te "uncommenten".

De output zou er als volgt uit moeten zien.

```
--- STAP 3 ---
Naam: Cruise Rocket
Prijs: €2.50
Instructie: Nederlandstalig=true, leeftijd=10, omschrijving=Niet in de hand houden
Legaal: false
Hoogte: 40.0 meter
Kleuren:
  ROOD: 50%
  GROEN: 25%
  BLAUW: 25%
```



```
--> FOUT: Onjuiste kleurverhouding, kleur wordt rood
Naam: Killing Arrow
  Prijs: €4.25
  Instructie: Nederlandstalig=true, leeftijd=16, omschrijving=Niet in de hand houden
  Legaal: true
  Hoogte: 40.0 meter
  Kleuren:
    ROOD: 100%
    GROEN: 0%
    BLAUW: 0%
--> FOUT: Onjuiste kleurverhouding, kleur wordt rood
Naam: Magic Sky
  Prijs: €2.75
  Instructie: Nederlandstalig=false, leeftijd=20, omschrijving=Keep minimum 10 ft distance
  Legaal: false
  Hoogte: 40.0 meter
  Kleuren:
    ROOD: 100%
    GROEN: 0%
    BLAUW: 0%
Naam: Golden Sky
  Prijs: €3.25
  Instructie: Ontbreekt
  Legaal: false
  Hoogte: 40.0 meter
  Kleuren:
    ROOD: 50%
    GROEN: 50%
    BLAUW: 0%
```

Deel II: Gebruik klassenstructuur

Stap 4: Lijst met vuurwerk (15 pt)

- Zorg dat stappen 1 tot en met 3 zijn uitgevoerd, want dan is de pakket-variabele gevuld met allerlei vuurwerk.
- Maak in de class `HappyNewYearDriver` een methode met de volgende signatuur:
`public static void toonVuurwerk(ArrayList<Vuurwerk> pakket)`
Deze methode moet alle soorten vuurwerk uit het pakket afdrukken. Aan het eind moet de totaalprijs (de som van de afzonderlijke prijzen van het vuurwerk) worden afgedrukt.
- Test je code door in de Driver alle code met TODO stap 4 te “uncommenten”.

De output zou er als volgt uit moeten zien.

```
--- STAP 4 ---
Naam: Veiligheidsbril
  Prijs: €2.50
  Instructie: Nederlandstalig=true, leeftijd=6, omschrijving=Draag bij aansteken
  Legaal: true
```



```
Naam: Safety glass
  Prijs: €2.50
  Instructie: Nederlandstalig=false, leeftijd=6, omschrijving=Wear before ignition
  Legaal: false

  ...
  (hier nog veel meer vuurwerk)
  ...

Naam: Golden Sky
  Prijs: €3.25
  Instructie: Ontbreekt
  Legaal: false
  hoogte: 40.0 meter
  kleuren:
    ROOD: 50%
    GROEN: 50%
    BLAUW: 0%

Kosten vuurwerkpakket: €240.5
```

Stap 5: Toon instructies (15 pt)

- Zorg dat stappen 1 tot en met 3 zijn uitgevoerd, want dan is de pakket-variabele gevuld met allerlei vuurwerk.
- Maak in de class `HappyNewYearDriver` een methode met de volgende signatuur:
`public static void toonInstructies(ArrayList<Vuurwerk> pakket, int index)`
Deze methode moet de instructie details afdrukken van het gekozen vuurwerk. Het gekozen vuurwerk wordt bepaald door de index.
Indien het uitgekozen vuurwerk geen instructie heeft moet de tekst "Instructie ontbreekt." worden afgedrukt. Indien de index buiten de grenzen van de lijst ligt moet de tekst "Index valt buiten grenzen." worden afgedrukt.
- Test je code door in de Driver class alle code met TODO stap 5 te "uncommenten".

De output zou er als volgt uit moeten zien.

```
--- STAP 5 ---
Index valt buiten grenzen.
Nederlandstalig=false, leeftijd=21, omschrijving=Keep minimum 10 ft distance
Instructie ontbreekt.
Index valt buiten grenzen.
```

Stap 6: Print harde knallers (10 pt)

- Zorg dat stappen 1 tot en met 3 zijn uitgevoerd, want dan is de pakket-variabele gevuld met allerlei vuurwerk.



- Maak in de class `HappyNewYearDriver` een methode met de volgende signatuur:
`public static void printHardeKnallers (ArrayList<Vuurwerk> pakket, int maxDecibel)`

Deze methode moet van de lijst alle knallers printen die harder zijn dan `maxDecibel`.

- Test je code door in de `Driver` class alle code met `TODO` stap 6 te “uncommenten”.

De output zou er als volgt uit moeten zien.

```
--- STAP 6 ---
Naam: Peking Rol
  Prijs: €45.00
  Instructie: Nederlandstalig=true, leeftijd=21, omschrijving=Houd minimaal 5 meter afstand
  Legaal: true
  aantalKnallen: 500
  decibel: 120
Naam: Shanghai Rol
  Prijs: €85.00
  Instructie: Nederlandstalig=true, leeftijd=21, omschrijving=Houd minimaal 5 meter afstand
  Legaal: false
  aantalKnallen: 1000
  decibel: 125
```