

**Eindopdracht module Java Fundamentals 1**

Inhoud

[Eindopdracht Java 3](#_Toc60663463)

[Maak een objectklasse Dobbelsteen 3](#_Toc60663464)

[Maak een driverklasse Dobbelsteentester 5](#_Toc60663465)

[Wat lever je in? 6](#_Toc60663466)

[Beoordeling 7](#_Toc60663467)

[Cesuur 9](#_Toc60663468)

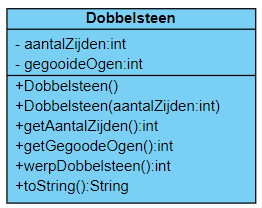
# Eindopdracht Java

Als eindopdracht maak je een applicatie met twee klassen:

* Dobbelsteen (objectklasse)
* Dobbelsteentester (driverklasse)

Een voorbeeld van de werking zie je op: <https://bit.ly/3rQG0lx>

# Maak een objectklasse Dobbelsteen



***Klassevariabelen:***

* aantalZijden à het aantal zijden dat de dobbelsteen heeft.
* gegooideOgen à het aantal ogen wat je met een aantal worpen hebt gegooid.

***Constructors:***

* Schrijf een standaard constructor die een standaard dobbelsteen met 6 zijden maakt.
* Schrijf een constructor met een parameter waarin je het aantal zijden kan meegeven zodat je ook dobbelstenen kan maken die een ander aantal zijden heeft.

***Getters en setters:***

Schrijf de getAantalZijden en de getGegooideOgen die beide een integer teruggeven.

***Methoden:***

werpDobbelsteen()

Deze methode geeft een integer terug. De methode genereert een waarde van 1 tot het aantal zijden van de dobbelsteen. Je kan hiervoor de methode nextInt() uit de klasse Random gebruiken.

Je moet daarvoor wel de import van java.util.Random doen. In de API documentatie kan je checken hoe e.e.a. werkt.

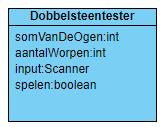
<https://docs.oracle.com/en/java/javase/13/docs/api/java.base/java/util/Random.html>

toString()

Schrijf een nieuwe implementatie van de toString() methode die de dobbelsteen laat weergeven.



# Maak een driverklasse Dobbelsteentester



Maak in de klasse het volgende scenario.

* Maak een dobbelsteen m.b.v. de standaard constructor.
* Toon het aantal zijden van deze dobbelsteen in je scherm
* Maak een dobbelsteen m.b.v. de constuctor met parameter en maak een dobbelsteen met 20 zijden.
* Toon het aantal zijden van deze dobbelsteen in je scherm
* Maak een lus(while) met daarin de volgende mogelijkheden
  + Met welke dobbelsteen wil je spelen? 1 of 2?
  + Hoe vaak wil je gooien?
  + Gooi het aantal keren met de dobbelsteen. Gebruik hiervoor een lus(for of while)
    - Print per worp de dobbelsteen
  + Toon de som van de ogen die gegooid zijn
  + Vraag of de speler nogmaals wil spelen
* Toon de boodschap bedankt voor het spelen bij het einde van het spel

# Wat lever je in?

Je levert op teams de volgende zaken in:

* Het bestand Dobbelsteen.java
* Het bestand Dobbelsteentester.java
* Een word bestand met een link naar een filmpje wat je op Microsoft Stream hebt gezet.  
  **Denk eraan dat je de docent als lezer toevoegt!!!!**
  + Het filmpje mag maximaal 5 minuten duren en moet de volgende onderdelen bevatten.
    - Een demo zoals je die hier ziet: <https://bit.ly/3rQG0lx>
    - Per klasse een korte uitleg over hoe je e.e.a. gedaan hebt.
    - TIP: maak een scriptje zodat je weet wat je moet zeggen. Een te lange video kost je punten.

# Beoordeling

| Onderdeel | 0 | 1 | 2 | 3 |
| --- | --- | --- | --- | --- |
| Naamgeving | Naamgeving is nergens volgens de afspraken. | De naamgeving van de variabelen, constructors, methoden en klassen voldoen voor minder dan 30% aan de daarvoor gestelde eisen. | De naamgeving van de variabelen, constructors, methoden en klassen voldoen voor minder dan 70% aan de daarvoor gestelde eisen. | De naamgeving van de variabelen, constructors, methoden en klassen voldoen aan de daarvoor gestelde eisen. |
| Commentaar | Er is geen commentaar aanwezig in de code. | Bij in ieder geval 70% van de code staat zinnig commentaar | Bij in ieder geval 50% van de code staat zinnig commentaar | Bij alle onderdelen in de code staat zinnig commentaar. |
| Werking van de code | De code werkt niet en bevat fouten waardoor niet gecompileerd kan worden. | Minder dan 50% van de code werkt naar behoren. De uitvoer op het scherm voldoet niet aan de eisen. | 75% van de code werkt en de uitvoer lijkt grotendeels op wat er in de demo te zien is. | De code werkt perfect en werkt zoals in de demo te zien is. |
| Constructors | De constructors zijn niet aanwezig of voldoen niet aan de eisen. | Alleen de standaard constructor werkt naar behoren. De andere constructor werkt niet goed of is niet aanwezig. |  | Beide constructors werken naar behoren. |
| Getters | De getters ontbreken. | De werking van de getters is niet correct of geeft niet de juiste waarde terug. |  | De werking van de getters is correct en de goede waarde wordt terug gegeven. |
| werpDobbelsteen() | De methode ontbreekt of geeft geen waarde terug | Berekening niet correct of geen correcte waarde. Werkt alleen met een standaard dobbelsteen. | Geeft een correcte waarde terug maar werkt alleen met de standaard dobbelsteen. | De methode geeft een juiste waarde gebaseerd op het aantal zijden van de dobbelsteen. |

| Onderdeel | 0 | 1 | 2 | 3 |
| --- | --- | --- | --- | --- |
| toString() | Er is geen nieuwe implementatie geschreven van toString(). | Er is een nieuwe implementatie van toString() geschreven maar deze voldoet niet aan de voorwaarden en/of geeft niet de juiste uitvoer en/of is niet correct uitgelijnd. | Er is een nieuwe implementatie van toString() geschreven maar deze is niet correct uitgelijnd. | Er is een nieuwe implementatie van toString() die werkt zoals in de demo. |
| Maken dobbelstenen | Het is niet gelukt de dobbelstenen te maken m.b.v. de constructors en het aantal zijden te tonen op het scherm. | Het is niet gelukt alle dobbelstenen te maken m.b.v. de constructors of het aantal zijden te tonen op het scherm. | De dobbelstenen zijn op correcte wijze gemaakt maar zijn niet getoond op het scherm. | Het is gelukt om de dobbelstenen te maken en ze te tonen op het scherm |
| Keuze | Het is niet mogelijk om te kiezen met welke dobbelsteen je gaat spelen | Het is wel mogelijk om te kiezen met welke dobbelsteen je gaat spelen |  |  |
| Herhaling | Het spel kan niet vaker dan 1 keer worden gespeeld. | Het spel kan vaker dan 1 keer worden gespeeld maar is niet geprogrammeerd m.b.v. een while loop. |  | Het spel kan vaker dan 1 keer worden gespeeld m.b.v. de while loop. |
| Aantal keren gooien | Het is niet mogelijk om aan te geven hoe vaak je de dobbelsteen wil gooien en de som van de ogen wordt niet of onjuist getoond. | Het niet mogelijk om aan te geven hoe vaak je wil gooien maar de som van de ogen wordt wel op correcte wijze getoond. | Het is mogelijk om vaker te gooien maar de som van de ogen wordt niet juist berekend en getoond. | Het is mogelijk om aan te geven hoe vaak je de dobbelsteen wil gooien en de som van de ogen wordt op correcte manier getoond |
| Inleveren | Alle bestanden zijn ingepakt in een zip-bestand. | De bestanden zijn los ingeleverd en zo makkelijk voor ons in te zien. |  |  |
| Video\* | Er is geen video aanwezig of deze is niet gedeeld met de docent of de video is langer dan 5 minuten. | De video is beschikbaar maar duurt langer dan 5 minuten. | De video is beschikbaar en de uitleg vindt plaats binnen de 5 minuten. |  |

\* Als de video niet aanwezig is of niet door ons te bekijken krijg je automatisch een N.

# Cesuur

Maximaal aantal punten = 34

|  |  |
| --- | --- |
| Aantal punten | Score |
| Niet ingeleverd of video niet beschikbaar | N |
| <19 | O |
| <27 | V |
| ≥27 | G |