



HBO – ICT

**TENTAMENVOORBLAD**

Voor aanvang van het tentamen s.v.p. het gehele tentamen goed doorlezen om eventuele misverstanden te voorkomen!

<b>Studieonderdeel</b>	:	Programming in Java
<b>Datum</b>	:	2 november 2020
<b>Tijd</b>	:	11.00 – 12.50 uur (+ 10 minuten uploaden)
<b>Propedeuse/Hoofdfase</b>	:	Propedeuse
<b>Leerroutes</b>	:	SE, TI en herkansers

<b>Aantal bladz. (incl.voorblad)</b>	:	5
<b>Toegestane hulpmiddelen</b>	:	<ul style="list-style-type: none"><li>• Boek van Liang (mag digitaal)</li><li>• IntelliJ &amp; Java Quick Reference (beschikbaar via TestVision)</li><li>• Uitwerkingen van je eigen practicumopdrachten en oefentoetsen</li></ul>
<b>Normering tentamen</b>	:	Zie toetstekst.
<b>Bijzonderheden</b>	:	Als je klaar bent met de opdracht, upload <u>het .java bestand</u> dan naar Testvision.

**Veel succes!**



# HvA - Among Us

---

## Tip vooraf

Bij deze toets maak je gebruik van een scanner om invoer in te lezen. Als je na een getal een string inleest, levert dit echter een onverwacht resultaat op: de gebruiker krijgt niet de kans om een string in te voeren. Dit komt omdat na het lezen van een getal de <Enter> nog in de buffer blijft staan. Als je daarna een string probeert te lezen dan wordt de <Enter> gezien en daardoor een lege string ingelezen. Dit probleem is te ondervangen door na het inlezen van een getal een `nextLine()` te doen om de buffer te legen. Dus stel dat de scanner invoer heet, dan voeg je na het inlezen van een getal het volgende statement toe: `invoer.nextLine();`

## Inleiding

Bouw het populaire spel 'Among us' na in Java! Het spel draait allemaal om het verraden van je vrienden! Je speelt dit spel met 3 tot 10 speler, deze geef je eerst allemaal op. Wanneer alle spelers zijn vastgelegd zal de computer één speler bepalen welke de bedrieger is, ook wel de 'impostor'. Aan de speler is het om te raden wie van alle deelnemende spelers de bedrieger is!

## Beschrijving

Schrijf een applicatie die aan het doel van de opdracht voldoet. Kijk hierbij goed naar het outputvoorbeeld op de volgende bladzijden en volg het onderstaande stappenplan.

### 0. Begin met het afdrukken van je naam, klas en studentnummer.

1. Vraag de gebruiker hoeveel spelers er meespelen. Het aantal spelers dat kan meespelen is van 3 tot 10.
2. Controleer of de ingevoerde hoeveelheid ook daadwerkelijk is toegestaan, wanneer er wel een foutief antwoord is ingevuld toon je de melding *"Dit is geen geldig aantal!"* en vraag je het opnieuw.
3. Maak een array aan waarin je de namen van spelers kunt opslaan.
4. Vraag bij elke speler een naam. Toon hierbij het nummer van de speler en geef daarna de mogelijkheid om een naam in te vullen.
5. Maak een methode welke de volgende structuur krijgt:  

```
public static int geefRandomImpostorIndex(int aantalSpelers)
```

Zorg ervoor dat deze methode een random getal teruggeeft tussen de 0 en het aantal spelers dat de gebruiker bij stap 1 heeft gekozen.
6. Bepaal nu wie de willekeurige bedrieger (impostor) is. Gebruik hiervoor de methode uit de vorige stap. Het random getal uit de methode verwijst in de array naar de index van de impostor.



7. Maak een methode welke de volgende structuur krijgt:  

```
public static void printSpelersNamen(String[] spelersLijst)
```

Deze methode print de namen van alle spelers uit. Elke naam begin je met het nummer van de speler zodat een gebruiker die kan invoeren. Gebruik voor de opmaak van de spelerslijst output `System.out.printf`.
8. Vraag de gebruiker om een nummer van een speler in te vullen.
9. Controleer of het ingevoerde nummer overeenkomt met de impostor. Wanneer het ingevoerde nummer niet overeenkomt met de impostor toon je de melding "[NAAM]<sup>1</sup> was niet de impostor. Probeer het nog eens." (zie de outputvoorbeelden op blad 4). Het programma geeft weer de lijst met namen en vraagt de gebruiker opnieuw om een nummer van een speler in te vullen. Dit herhaalt zich zolang de impostor niet juist is geraden.
10. Wanneer het ingevoerde nummer overeenkomt met de impostor toon je de melding "Goed geraden! [NAAM]<sup>1</sup> was inderdaad de impostor." (zie de outputvoorbeelden op blad 4). Je hoeft niet te controleren of het ingevoerde getal wel geldig is.
11. Als de gebruiker de impostor goed heeft geraden geeft het programma aan hoeveel keer de speler nodig had om de impostor te ontmaskeren en stopt het programma.

<sup>1</sup> [NAAM] vervangen door de naam welke hoort bij het ingevoerde nummer.

## Richtlijnen bij coderen (zie ook HBO-ICT code conventions)

- Zorg dat je naam en het doel van het programma bovenin staan.
- Gebruik de juiste inspringing (*indentation*) bij de lay-out.
- Let op juist gebruik hoofdletters en kleine letters.
- Gebruik goede namen.
- Voorkom Magic Numbers.
- Voeg waar nodig commentaar toe dat inzicht geeft in je code.



## Outputvoorbeelden

Zorg dat de output van je programma er precies hetzelfde uitziet als onderstaand outputvoorbeelden.

Voor de duidelijkheid zijn in het voorbeeld de gegevens die de gebruiker heeft ingevoerd schuin en onderstreept. Jouw programma hoeft dus niets schuin en onderstreept af te drukken.

```
Dit programma is gemaakt door <student>, <klas>, <studentnummer>
```

```
Hoeveel spelers doen er mee [3 - 10]? 4
```

```
Vul per speler een naam in.
```

```
Speler 1: Teresa
```

```
Speler 2: Selcuk
```

```
Speler 3: Zhen
```

```
Speler 4: Mats
```

```
Raad nu de impostor!
```

1. Teresa
2. Selcuk
3. Zhen
4. Mats

```
Vul het nummer van de speler in: 1
```

```
Teresa was niet de impostor. Probeer het nog eens.
```

1. Teresa
2. Selcuk
3. Zhen
4. Mats

```
Vul het nummer van de speler in: 3
```

```
Goed geraden! Zhen was inderdaad de impostor.
```

```
Je had 2 keer nodig om de impostor te ontmaskeren.
```

```
Dit programma is gemaakt door <student>, <klas>, <studentnummer>
```

```
Hoeveel spelers doen er mee [3 - 10]? 2
```

```
Dit is geen geldig aantal!
```

```
Hoeveel spelers doen er mee [3 - 10]? 3
```

```
Vul per speler een naam in.
```

```
Speler 1: Henk
```

```
Speler 2: Adnan
```

```
Speler 3: Barry
```

```
Raad nu de impostor!
```

1. Henk
2. Adnan
3. Barry

```
Vul het nummer van de speler in: 1
```

```
Goed geraden! Henk was inderdaad de impostor.
```

```
Je had 1 keer nodig om de impostor te ontmaskeren
```



## Normering

- Als het programma niet compileert heb je een 1.
- Als het programma wel compileert, zal het cijfer worden bepaald door “onder de motorkap” te kijken en te letten op het ontwerp van het programma, inclusief het gebruik van de juiste datatypes, een methode, arrays en het voldoen aan richtlijnen bij coderen. Hierbij zijn de maximaal te behalen punten per onderdeel als volgt:
  - Stap 1: Inlezen aantal spelers: **5 pt**
  - Stap 2: Test op foutieve invoer en indien nodig herhaaldelijk om nieuwe invoer vragen: **10 pt**
  - Stap 3: Aanmaken van array met correct datatype en lengte: **5 pt**
  - Stap 4: Invoeren van spelersnamen met correct volgnummer: **10 pt**
  - Stap 5 en 6: Aanmaken van het random impostor getal: **15 pt**
    - Aanmaken en aanroepen van methode met juiste signatuur: 10 pt
    - Random genereren van getal: 5 pt
  - Stap 7: Printen van namen van de spelers: **15 pt**
    - Aanmaken en aanroepen van methode met juiste signatuur: 5 pt
    - Printen van namen met correct volgnummer met gebruik van printf: 10 pt
  - Stap 8, 9 en 10 Raden van de impostor: **20 pt**
    - Keuze speler verwerken en melding afdrukken indien fout geraden en herhalen: 15 pt
    - Keuze speler verwerken en melding afdrukken indien goed geraden en afbreken: 5 pt
  - Stap 11: Printen hoeveel keer de speler nodig had om de impostor te ontmaskeren: **10 pt**
  - Voldoen aan de codeerrichtlijnen: **10 pt**

Cijfer: aantal behaalde punten / 10.