

**Het Arsenaal, een oude opslagplaats van militaire goederen midden in het centrum van Woerden, is het decor van de vijfde editie van Masters of Java. Binnen in het uit 1762 daterende munitiedepot is het net alsof je een stap in het verleden hebt gedaan. De twintig deelnemers aan Masters of Java hebben echter weinig oog voor de historische setting. Zij vechten hun eigen ‘oorlog’ uit met wapens die in de tijd dat het gebouw nog voor militaire doeleinden werd gebruikt totaal niet voorhanden waren. Ze kruisen de degenen via hun laptops om een gooi te doen naar de titel.**

# Masters of Java 2008

## Speelse Java-competitie in historische setting

**D**oodstil is het in de zaal. De 25 deelnemers aan deze editie van Masters of Java, verdeeld over elf teams, staren geconcentreerd naar de laptop die zich voor hen op de tafel bevindt. Soms volgt er overleg en dan even het getik op een toetsenbord, gevolgd door een blik naar het grote scherm om het resultaat te bewonderen. De stilte wordt soms onderbroken door een triomfantelijk gebaar, gebalde vuist of een krachtterm van vreugde wanneer een opgave is opgelost. Maar ook de teleurstelling en lichte frustratie is voelbaar bij de teams wanneer de tijd om is en het probleem niet is opgelost.

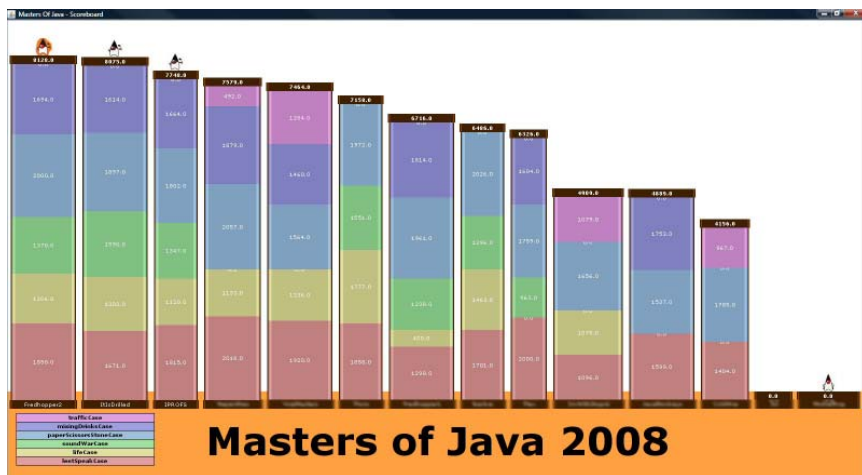
Een half uur hebben ze tijd om een opgave op te lossen. Zoals inmiddels vertrouwd, zijn deze weer uitgedacht door Erik Hooijmeijer van het bedrijf ‘42’, die er weer in geslaagd is grappige scenario’s te koppelen aan de benodigde kennis van Java onder de deelnemers. “Er zit weer van alles in. Hoewel de context van de opgaven grappig van toon is, testen we de deelnemers op hun kennis van Java. Sommige opgaven zijn met goede basiskennis van Java op te lossen, terwijl de moeilijkere puzzels meer vergen. Mooie uitdagingen dus”, vindt de bedenker van de opgaven. De competitie bestaat uit zes ronden van een half uur. De deelnemers programmeren tot aan de testfase en moeten ervoor zorgen dat alle tests per opgave correct kunnen worden uitgevoerd. Pas wanneer alle tests groen licht opleveren, kunnen er punten worden gescoord. Een volledig opgeloste opdracht levert vierhonderd punten op. Daarnaast vallen er ook punten te verdienen voor de tijd die de teams ongebruikt laten. Elke seconde die nog op de klok staat wanneer de oplossing wordt ingediend, levert een punt op. Maximaal zijn er dus 2200 punten per opgave te verdelen.

Slagen niet alle tests, dan kunnen er geen punten worden toegekend.

Onverbidlijk tikt de klok voor in de zaal weg. Al snel ontstaat bij iedere opdracht een tweedeling tussen de deelnemers. De teams die al snel het probleem hebben opgelost zitten ontspannen achter hun laptops, terwijl de anderen nog zitten te ploeteren en proberen te ontdekken waar het mis is gegaan. “Dat is altijd wel leuk om te zien. Aan de ene kant de mensen die al klaar zijn en dolgraag hun oplossing willen prijsgeven en daarnaast degenen die er maar niet uitkomen en steeds gefrustreerder achter de tafel gaan zitten”, heeft Hooijmeijer ervaren.

Na elke ronde volgt een korte evaluatie. Het team dat als eerste klaar is met de opgave mag dan uit de doeken doen hoe ze het gedaan hebben. Vaak gaan deze evaluaties gepaard met herkenning uit de zaal en balende gezichten van de teams die ergens onderweg in de opgave iets gemist hebben of vergeten zijn.

Na een testronde om de deelnemers te laten wennen aan het systeem volgt dan daadwerkelijk de aftrap van Masters of Java 2008. Ondanks de matige bezetting – slechts elf van de ingeschreven dertien equipes verschijnen aan het vertrek – is het ongemeen spannend. Vijf teams zijn na de ochtendsessie van vier opdrachten nog volop in de race voor de eindoverwinning. De beslissing valt na de lunchpauze bij de voorlaatste opgave. Daar legt het team Fredhopper 2 van Bram Bouwes en Marc Marschner de basis voor hun overwinning. Zij weten het beste raad met de case over de logistieke problemen die het drinken van Wodka-Jus met zich mee kan brengen. Zij vinden het snelst de oplossing voor het probleem wie een nieuwe voorraad drank mixt als die op is.



De einduitslag van Masters of Java 2008.

De laatste en moeilijkste opgave zorgt nog voor enige hilariteit. Na een paar minuten springen alle testschermen van team Masters of Sogyo al op groen. De andere teams ploeteren nog voort op de lastige kwestie van een parkeerwachter bij een groot restaurant. Pas in de laatste minuten slaagt ook een aantal andere teams erin de opgave op te lossen. Wat blijkt bij de evaluatie: er zijn twee manieren om het probleem te tackelen. Er is de normale weg via de gehele code, maar er blijkt ook via een achterdeur een weg naar de juiste oplossing. Die hebben de Masters of Sogyo gevonden en zo hun concurrenten afgetroefd.

De eindoverwinning levert het zoals eerder gezegd niet op. Die is voor Fredhopper 2. Zij weten 'ItIsDrilles' achter zich te houden. De derde plaats is voor Team IPROFS.

### Een opgave uitgelicht

Zes opgaven kregen de deelnemers voorgeschoteld. Een daarvan was gebouwd rondom het welbekende spelletje 'Paper, Scissors, Stone'. Van deze case doen we de oplossing uit de doeken. Maar eerst even een korte introductie zoals de deelnemers die ook voorgeschoteld kregen.

**Paper, Scissors, Stone** – De laatste jaren is het aantal 'easy to play games' enorm toegenomen. Ook jouw bedrijf wil zich mengen in deze markt. Om dit soort games te maken, moet je terug naar de basis. Je kunt niets beter dan een simpel spelletje als 'Paper, Scissors, Stone' als basis nemen. Dit is ook het uitgangspunt van menig strategisch spel. Het geld stroomt dus al bijna binnen. Alleen moet er nog een programmeur gevonden worden die het spel maakt.

Implementeer de battles zo in de enumeratie 'PaperScissorsStoneImpl' dat de regels van het spel gevolgd worden. De regels van het spel zijn:

- Papier is sterker dan steen (papier pakt steen in)
- Steen is sterker dan schaar (schaar wordt bot)
- Schaar is sterker dan papier (knipt het door)

Als twee gelijken tegen elkaar strijden is de uitkomst onbeslist.

Voor deze opgave zijn drie mogelijke manieren te bedenken om hem op te lossen. Tijdens Masters of Java 2008 imponeerde het winnende team van deze opdracht door een nog kortere implementatie te schrijven. Het onderwerp van de Paper Scissors Stone case is het gebruik van enumeraties, een feature waar nog veel Java-programmeurs een beetje aan moeten wennen.

```
public enum PaperScissorsStoneImpl {
    PAPER(2), SCISSORS(0), STONE(1);

    private int win;

    private PaperScissorsStoneImpl(int win) {
        this.win=win;
    }

    public Outcome battles(PaperScissorsStoneImpl
other) {
        if (other.ordinal()==win) return Outcome.
WIN;
        if (other.equals(this)) return Outcome.TIE;
        return Outcome.LOSE;
    }
}
```

Om deze opgave tot een goed einde te brengen, zijn meerdere oplossingen te vinden. De volgende uitwerking is gebaseerd op hard werken.

```
public Outcome battles(PaperScissorsStoneImpl
other) {
    switch (this) {
        case PAPER:
            if (STONE.equals(other)) return Outcome.
WIN;
            if (SCISSORS.equals(other)) return Outcome.
LOSE;
            break;
        case SCISSORS:
            if (PAPER.equals(other)) return Outcome.
WIN;
            if (STONE.equals(other)) return Outcome.
LOSE;
            break;
        case STONE:
            if (SCISSORS.equals(other)) return Outcome.
WIN;
            if (PAPER.equals(other)) return Outcome.
LOSE;
            break;
    }
    return Outcome.TIE;
}
```

Zoals al eerder gezegd, zijn er meerdere wegen die naar Rome leiden. De echte slimmerikken kunnen dit probleem ook via de volgende oplossing tackelen. «

```
public Outcome battles(PaperScissorsStoneImpl
other) {
    int delta=this.ordinal()-other.ordinal();

    if (delta==0) return Outcome.TIE;

    if ((delta==1)|| (delta==2)) return Outcome.
LOSE;

    return Outcome.WIN;
}
}
```