

Bij de volgende opgaven maken wij gebruik van opgegeven waarden via de commandline. Hierbij dient het script uitgevoerd te worden vanuit een MS-DOS (of Linux) commandline. De gevraagde waarden kunnen na de filename worden ingevoerd, op deze manier:

```
groovy mijnScript invoer1 invoer2
```

Het uitlezen van deze waarden kan in het script vervolgens middels:

```
println args[0] // invoer1
println args[1] // invoer2
```

Als je een args waarde wilt omzetten naar een getal (want het is standaard een String), gebruik dan:

```
int getal = invoer1 as Integer
of:
int getal = Integer.parseInt(invoer1)
```

Programmeerstructuren

- Schrijf een script dat de hoogste waarde vindt van twee ingevoerde getallen vindt en toont in de console
- Schrijf een script dat bij een opgegeven aankoop- en een opgegeven verkoopprijs een melding in de console toont of er sprake is van winst of verlies
- Schrijf een script dat van een opgegeven weekday aangeeft of het een werkdag of een weekenddag is
- Toon een optelling van alle getallen van 1 t/m 100
- Bereken alle priemgetallen vanaf 3 t/m een opgegeven getal. Priemgetallen zijn getallen die enkel en alleen deelbaar zijn (er na deling 0 overblijft) door zichzelf en door 1. Bijvoorbeeld het getal 5 of 7. Het de getallen 6, 8 en 9 zijn al deelbaar door respectievelijk 2, 4 en 3.

Collecties

- Maak een list met je favoriete gerechten
 - Voeg een gerecht toe aan de list
 - Haal het laatste element uit de list
 - Haal het eerste element uit de list
 - Toon met een methode of de list leeg is
 - Toon het aantal elementen dat in de list aanwezig is
- Maak de volgende map aan in een Groovy script:
def consoles = [nintendo: 'NES', sega: 'Master System', atari: '2600']
 - Toon alle keys uit deze map. Hint: maak hiervoor gebruik van een for-in loop door de keySet() methode van deze map
 - Toon nu door dezelfde loop te gebruiken alle values van de map

Mini Casus: Producten beheer (2)

Wij gaan verder met onze mini casus.

Bonuspunten: begin met een verzameling van Strings, maar je kan dit als je aan het **einde van de cursus** tijd over hebt uitbreiden naar objecten.

1) Vraag de eindgebruiker om telkens een keuze te maken wat hij of zij wilt. Dit kan door gebruik te maken van de `System.in.withReader ()` functie:

```
System.in.withReader ({
    console -> {
        println "Geef je keuze op:"
        ...
    }
})
```

Binnen deze functie kun je de gebruiker een keuze geven uit vier opties:

- 1) Toon producten
- 2) Voeg product toe
- 3) Wijzig product
- 4) Verwijder product

Doe dit in een loop, omdat anders de applicatie al klaar is na de eerste keus.

Zorg ervoor dat er in de loop wordt afgevangen wanneer er bijvoorbeeld op 'q' wordt gedrukt om de applicatie mee te sluiten:

```
def invoer = console.readLine()

while (invoer != 'q') {
    println "Je keuze was: ${invoer}";
    println "Geef je keuze op:"
    ...
    invoer = console.readLine();
}
```

Gebruik de regel: `invoer = console.readLine()` om telkens een nieuwe waarde te laten lezen wat de eindgebruiker wilt doen.

We mogen zelf opzoeken welke methoden van een list wij nodig gaan hebben. Deze zijn hier te vinden:

<http://docs.groovy-lang.org/latest/html/groovy-jdk/java/util/List.html>

Voor de toevoeg, wijzig en verwijder functionaliteit mogen wij voor verwijzen naar onze eerder aangemaakte methoden

Tijd over? Breid onze methoden met bijvoorbeeld een controle op soort gegeven dat er binnen komt correct is, zoals m.b.v.: [instanceof](#)

Succes!

Objectoriëntatie; Klassen, referentie variabelen en modifiers

- Maak een klasse van een voor jou aansprekend artikel. Zorg dat deze klasse minimaal bevat:
 - Vier eigenschappen
 - Een methode die een argument(en) accepteert als input
 - Een methode met een return waarde
- Maak een tweetal instantie van deze klasse aan vanuit je script
- Toon van beide instanties een nuttige representatie in de console. Hint: overschrijf hiervoor de `toString()` methode van de klasse `Object`
- Zorg ervoor dat bij een vergelijking van deze twee instanties met de `equals` methode of `==`, de vergelijking `true` oplevert indien twee van de zelf gekozen eigenschappen gelijk aan elkaar zijn
- Maak zelf een voorbeeld waarbij je het verschil kunt aantonen tussen `pass-by-reference` en `pass-by-value`
- Bedenk of je een attribuut aan je klasse kunt toevoegen waaraan je de `static` modifier kunt toevoegen.

Testing

- Probeer de regels van het bestand: `leesmij.txt` in te lezen en op je scherm te tonen. Maak hiervoor gebruik van de File klasse. Kijk hier een voorbeeld: https://www.tutorialspoint.com/groovy/groovy_file_io.htm
- Schrijf een script waarbij wij middels een try/catch systeem proberen een tweetal getallen te vermenigvuldigen.
 - Indien de via de commandline meegegeven waarden geen getallen zijn (NumberFormatException) of er zijn niet voldoende argumenten meegegeven (ArrayIndexOutOfBoundsException) dienen deze fouten netjes met een melding afgehandeld te worden.
 - Indien het goed gaat mag het resultaat van de vermenigvuldiging in de console getoond worden.
- Maak een klasse 'Brommer' met een eigenschap tankInhoud en een methode rijden(). De methode rijden() dient de waarde van de tankInhoud te verlagen met 1.
 - Test de werking van deze klasse door in een script een assert toe te passen
 - Test de werking van deze klasse door een Testklasse te maken en een Groovy Test Suite (op de manier zoals behandeld in de sheets en/of op: https://www.tutorialspoint.com/groovy/groovy_unit_testing.htm)

Mini Casus: Producten beheer (2)

- Voeg een nieuwe optie aan je Casus-menu toe om de collectie te persisteren (weg te schrijven) naar een bestand. Je kunt hiervoor eventueel de volgende naslag gebruiken: http://docs.groovy-lang.org/latest/html/documentation/working-with-io.html#_data_and_objects
- Probeer tot slot ook of je als extra menu optie het bestand kunt laten inlezen als vervanging van je bestaande collectie.

Succes!

Libraries en tools

- Probeer de besproken tools te installeren en een demo project mee te starten.
 - Dit hoeft niet from scratch, maar je mag hiervoor per project het aangegeven voorbeeld gebruiken om een beeld te krijgen van de opbouw en omvang.