

Java Fundamentals - Labo 8

Richtlijnen

- Onderstaande oefeningen worden in de Netbeans IDE uitgevoerd. Denk aan de debug mogelijkheden zoals inge oefend tijdens vorig labo !
- Maak 1 project aan (met als naam Labo8), voor elke nieuwe opgave maak je een nieuwe package aan binnen dit project.
- Wanneer je een iteratie moet schrijven kies jezelf welke lus het meest geschikt is, tenzij expliciet gevraagd.

1 Combinaties van iteraties en selecties

1. **ForToWhile** Herschrijf volgend codefragmentje zodat geneste while statements gebruikt worden ipv de for-lussen :

```
int s = 0;
int t = 1;
for (int i = 0; i < 5; i++) {
    s = s + i;
    System.out.println("S is " + s);

    for (int j = i; j > 0; j--) {
        t = t * (i - j + 1);
    }

    System.out.println("T is " + t);
}

s = s * t;
System.out.println("S is " + s);
```

2. **PositiefVershil** Schrijf een programma dat een waarde voor n inleest, gevolgd door n getallen, en dat het positieve verschil tussen het grootste en het kleinste getal van deze reeks uitschrijft.
3. **PascalToJava** Schrijf een programma dat een tekst inleest en weer uitschrijft, maar daarbij telkens het woord "Pascal" vervangt door het woord "Java". Controleer of je eigen implementatie hetzelfde resultaat oplevert als de `replaceAll` methode van de klasse `String`.
4. **Palindroom** Schrijf een programma dat voor een ingegeven woord bepaalt of het een palindroom is of niet. Een palindroom is een woord dat gelijk blijft als je het omgekeerd leest, bv. *lepel* of *meetsysteem*. Breid je programma vervolgens uit zodat je voor meerdere woorden de palindroomtest kan uitvoeren. Het programma stopt wanneer je geen woord meer ingeeft.

```
Geef een woord in: test
Geen palindroom
Geef een woord in: lepel
Palindroom
```

5. **PlusMinSom** Schrijf een programma dat een getal n inleest en vervolgens het resultaat van volgende som s bepaalt:

$$s = 1 - 2 + 3 - 4 + 5 - \dots \pm n$$

Bv. voor $n = 3$ is $s = 2$ en voor $n = 7$ is $s = 4$.

6. **Fibonacci** Lees een geheel getal n in tussen 1 en 20. Indien het getal niet tussen 1 en 20 ligt geef je een foutmelding en onderbreek je het programma. Bij correcte input schrijf je de Fibonacci rij uit tot het n -de Fibonacci-getal. De Fibonacci-getallen definieer je als volgt :

$$F_1 = 1 \quad F_2 = 1 \quad F_3 = 2 \quad F_4 = 3 \quad F_5 = 5 \quad F_6 = 8 \quad F_7 = 13 \quad \dots$$

m.a.w. $F_n = F_{n-1} + F_{n-2}$ (op het eerste en het tweede Fibonacci-getal na is elk Fibonacci-getal de som van de twee voorgaande Fibonacci-getallen).

7. **TafelsVanVermenigvuldiging** Schrijf een programma dat een getal inleest en via iteratie de tafels van vermenigvuldiging geeft voor alle getallen kleiner of gelijk aan dit getal. Bijvoorbeeld voor de waarde 6 verschijnt de volgende tabel op het scherm :

1	2	3	4	5	6
2	4	6	8	10	12
3	6	9	12	15	18
4	8	12	16	20	24
5	10	15	20	25	30

8. **BladSteenSchaar** Implementeer het spelletje blad-steen-schaar. Twee spelers kiezen tesamen voor blad, steen of schaar. Blad wint van steen, steen wint van schaar en schaar wint van blad. Als beide spelers hetzelfde kiezen is het gelijkspel. Schrijf een programma waarbij je dit spel tegen de computer kunt spelen. Zorg ervoor dat je programma blijft vragen naar jouw keuze tot je een geldige ingave doet: nl. blad, steen of schaar.

```
Maak je keuze: Blad, Steen of Schaar
test
Maak je keuze: Blad, Steen of Schaar
Blad
Speler koos BLAD - Computer koos STEEN => Jij ➡
➡hebt gewonnen!
Druk op een toets om door te gaan. . .
```

9. **Dobbelen** Schrijf een programma dat het werpen van een dobbelsteen simuleert, en dat telt hoeveel worpen nodig zijn vooraleer een gegeven aantal ogen geworpen wordt. Print telkens uit hoeveel ogen er in elke poging geworpen werd, bijvoorbeeld:

```
Hoeveel ogen wil je werpen?
6
Poging 1 : 2
Poging 2 : 4
Poging 3 : 5
Poging 4 : 6
In 4 pogingen werd een 6 gedobbeld
```

10. **Model** Herneem oefening 11 van labo 3. Werk nu de waarheidstabel uit via iteratie.

Beschouw de volgende logische uitspraken :

uitspraak 1 Jan komt als Marie of Anne komt. Omgezet naar propositielogica geeft dit : $((m \vee a) \rightarrow j)$ wat logisch equivalent is met : $\neg(m \vee a) \vee j$

uitspraak 2 Anne komt als Marie niet komt, oftewel $\neg m \rightarrow a$ wat logisch equivalent is met $m \vee a$

uitspraak 3 Jan komt niet als Anne komt, oftewel $a \rightarrow \neg j$ wat logisch equivalent is met $\neg(a \wedge j)$

Kunnen we hier uit afleiden wie komt en wie niet? Aan de hand van logische waardentabellen kunnen we een waardering zoeken die deze 3 uitspraken waar maakt. Schrijf hiertoe een programma dat volgende waardentabel berekent en toont op het scherm. Gebruik de logische operatoren om de waarden uit te rekenen. Voor welke logische waarden van m , a en j zijn de uitspraken 1, 2 en 3 tegelijkertijd waar? M.a.w. voor welke setting van waarderingen vind je een rij in je tabel terug die 3x true oplevert.

waarderingen			uitspraken		
m	a	j	1	2	3
true	true	true	true	true	false
true	true	false	false	true	true
true	false	true			
true	false	false			
false	true	true		...	
false	true	false			
false	false	true			
false	false	false			